

## Backup, restoration, migration systems of a database.

Patent Number:  EP0614150, A3, B1

Publication date: 1994-09-07

Inventor(s): FUSHIMI SHINYA C O MITSUBISHI (JP)

Applicant(s): MITSUBISHI ELECTRIC CORP (JP)

Requested Patent:  JP6259301

Application Number: EP19940103254 19940303

Priority Number(s): JP19930043850 19930304

IPC Classification: G06F15/40

EC Classification: G06F11/14A4B1M

Equivalents: DE69427381D, DE69427381T, JP2783109B2, KR9704101,  US5642505

Cited Documents:

---

### Abstract

---

This invention provides an easy backup, restoration and migration method of a database system. The command interpreter receives a backup request and the controller identifies the backup object in the backup request by referring to the definition. The backup-migrate-restore-procedure generator generates a backup procedure of the backup object identified by the controller by referring to the definition. The controller converts the backup object to a command form of SQL in accordance with the procedure generated by the backup-migrate-restore-procedure generator and stores in the storage medium. The stored data are the replace information, definition, dependence information, data, journal and program. Restoration of the system and migration of the database to another system is enable by inputting such data. Backup, migration and restoration of a system is able to be made in an identical apparatus. Furthermore, since a command form of SQL is used, a special means is unnecessary for migration and restoration.

---

Data supplied from the **esp@cenet** database - I2

---

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-259301

(43)公開日 平成6年(1994)9月16日

(51)Int.Cl.  
G 0 6 F 12/00

識別記号 庁内整理番号  
5 3 1 R 8944-5B

F I

技術表示箇所

(21)出願番号 特願平5-43850  
(22)出願日 平成5年(1993)3月4日

審査請求 未請求 請求項の数17 O L (全 53 頁)

(71)出願人 000006013  
三菱電機株式会社  
東京都千代田区丸の内二丁目2番3号  
(72)発明者 伏見 信也  
鎌倉市大船五丁目1番1号 三菱電機株式  
会社情報電子研究所内  
(74)代理人 弁理士 高田 守

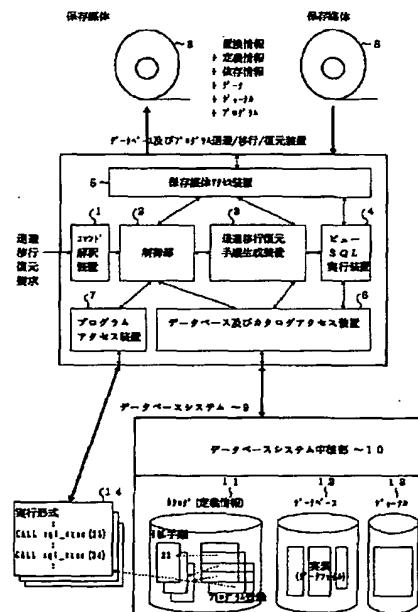
(54)【発明の名称】 データベースシステム退避装置及びデータベースシステム復元装置及びデータベースシステム移行装置

(57)【要約】

【目的】 データベースシステムの退避と復元移行を容易に行えるようにする。

【構成】 コマンド解釈装置1が、退避要求を受け付け、制御部2が退避要求から退避すべき対象物を定義情報11を参照することにより特定する。退避移行復元手順生成装置3は制御部で特定された退避すべき対象物の退避手順を定義情報11を参照しながら生成する。制御部は退避移行復元手順生成装置により生成された手順に従って退避すべき対象物を、SQLのコマンド形式に変換して保存媒体8に保存する。保存データは置換情報、定義情報、依存情報、データ、ジャーナル、プログラムからなり、これらを入力する事によりシステムの復元、あるいは他のシステムへのデータベースの移行が可能になる。

【効果】 システムの退避と移行と復元が同一装置により行える。またSQLのコマンド形式を用いるため、移行復元が特別な手段を必要とせずに行える。



## 【特許請求の範囲】

【請求項1】 以下の要素を有し、定義情報とこの定義情報により扱われるデータを記憶するデータベースシステムを退避するデータベースシステム退避装置

(a) 上記データベースシステムをアクセスするデータベースアクセス手段、(b) 退避すべきデータを保存データとして保存媒体に保存する保存媒体アクセス手段、(c) 上記データベースシステムの退避命令を入力して解釈するコマンド解釈手段、(d) 上記データベースアクセス手段によりデータベースシステムの定義情報をアクセスすることにより、退避すべき対象物の退避すべき手順を生成する退避手順生成手段、(e) 上記コマンド解釈手段により解釈された退避命令に基づき、上記データベースシステムの中から退避すべき対象物を特定するとともに、上記退避手順生成手段を動作させ、退避手順生成手段により生成された退避すべき手順に従って、退避すべき対象物を上記データベースアクセス手段によりデータベースシステムから取り出し、上記保存媒体アクセス手段により保存媒体に保存する制御手段。

【請求項2】 上記コマンド解釈手段は、退避する対象物として、少なくともユーザ名、SQLのスキーマなどのデータの分類記述単位名、SQLの表やファイルなどのデータ名、SQLのビューなどの論理的データ名、プログラム名、権限名のいずれかひとつの対象物の指定と、この対象物に対する種類と範囲の指定がなされた退避命令を入力するとともに、上記制御手段は、退避命令による対象物の指定とこの対象物に対する種類と範囲の指定に基づいて、定義情報を参照して指定された種類と、範囲に含まれる対象物を特定することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項3】 上記データベースシステムは、データベース処理言語を用いてアクセスでき、上記データベースアクセス手段は、データベース処理言語を用いて上記データベースシステムにアクセスすることを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項4】 上記制御手段は、保存媒体に保存する情報をデータベース処理言語が用いるコマンドの記述形式に変換して保存することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項5】 上記制御手段は、保存媒体に退避すべき対象物として定義情報を保存する場合に、保存する定義情報をデータベース処理言語による定義情報生成のための定義コマンドに変換して、保存媒体に保存することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項6】 上記制御手段は、保存媒体に退避すべき対象物としてデータを保存する場合に、保存するデータをデータベース処理言語によるデータ挿入のためのデータ挿入コマンドに変換して、保存媒体に保存することを特徴とする請求項1記載のデータベースシステム退避装置。

置。

【請求項7】 上記制御手段は、保存媒体に退避すべき対象物の中に置換可能な変数がある場合、その変数の値を置換可能にする置換情報を生成し、これを保存媒体に保存することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項8】 上記データベースシステムの定義情報は、データベース処理言語による実行形式の処理手順を有しており、上記データベースシステムは、さらに、上記処理手順のソースコードを保持するとともに上記定義情報にある実行形式の処理手順を実行してデータを処理するプログラムを有しており、上記データベースシステム退避手段は、さらに上記プログラムをアクセスするプログラムアクセス手段を有しており、上記制御手段は、上記定義情報にある実行形式の処理手順が退避すべき対象物になった場合でもこれを退避すべき対象物とはせず、代わりに退避すべき処理手順のソースコードを有するプログラムを退避すべき対象物とし、上記プログラムアクセス手段を用いて退避すべきプログラムをアクセスして取り出し、保存媒体に保存することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項9】 上記制御手段は、データベースのデータ記憶領域を確保するためのデータベース処理言語による領域確保コマンドを生成して保存することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項10】 上記データベースシステムは、さらに、データベースへのアクセス履歴情報をデータの識別子とともに記録するジャーナルを有し、上記制御手段は、上記ジャーナルから識別子を検索することにより、保存媒体に退避すべき対象物に関連するデータのアクセス履歴情報を選択し、これを退避すべきジャーナルとして、保存媒体に保存することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項11】 上記退避手順生成手段は、退避すべき手順を生成する際に、退避すべき対象物間の依存関係を特定し、上記制御手段は、上記退避手順生成手段が特定した依存関係を依存情報として保存することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項12】 上記請求項1記載のデータベースシステム退避装置により保存媒体に保存された保存データを入力し、上記保存データに基づいてデータベースシステムを復元することを特徴とするデータベースシステム復元装置。

【請求項13】 上記請求項1記載のデータベースシステム退避装置により保存媒体に保存された保存データを入力し、上記保存データに基づいてデータベースシステムを新たに構築するデータベースシステム移行装置。

【請求項14】 上記請求項4、5、6記載のデータベースシステム退避装置により、データベース処理言語のコマンドの記述形式で保存媒体に保存された保存データ

を入力し、データベース処理言語により記述されたコマンドを実行することにより、データベースを復元又は移行することを特徴とするデータベースシステム復元装置又はデータベースシステム移行装置。

【請求項15】上記請求項7記載のデータベースシステム退避装置により保存媒体に保存された保存データを入力し、置換可能な変数に新たな変数の値を与え、置換情報に基づいて置換可能な変数を新たな変数の値に置き換えながらデータベースシステムを新たに構築することを特徴とするデータベースシステム移行装置。

【請求項16】上記請求項8記載のデータベースシステム退避装置により保存媒体に保存された保存データを入力し、データベースシステムを新たに構築するとともに、プログラムの実行時に、定義情報に実行形式の処理手順が存在しない場合、プログラムにある処理手順のソースコードをコンパイルして実行形式の処理手順を新たに作成することを特徴とするデータベースシステム移行装置。

【請求項17】上記請求項11記載のデータベースシステム退避装置により保存媒体に保存された保存データを入力するとともに、復元仕様又は移行仕様を入力し、上記保存データの依存関係を参照することにより、復元又は移行するデータを復元仕様又は移行仕様に適合するように評価して、データベースを復元又は移行する復元手順生成手段又は移行手順生成手段を有することを特徴とするデータベースシステム復元装置又はデータベースシステム移行装置。

#### 【発明の詳細な説明】

##### 【0001】

【産業上の利用分野】この発明は、データベースシステムのプログラム及びデータの退避、復元、移行方式に関するものである。

##### 【0002】

【従来の技術】関係データベースが急速に普及しつつある。そこでは、データベースの処理言語としてSQL (ISO/IEC 9075: Database Language SQL) と呼ばれる国際標準言語が用いられることが一般的である。このため、以下では関係データベースと国際標準データベース処理言語SQLを前提として説明する。

【0003】業務プログラムとデータベースの基本構成を図15に示す。一般にデータベースシステムは、処理対象のデータそのものに加えて、データの内容を記述したデータベースの定義情報と、これら定義情報を参照しながら動作する業務プログラムとから構成される。これら定義情報はカタログ、リポジトリ、データディレクトリ/ディクショナリ等と呼ばれる。また、データに対しては、常時更新が行なわれており、これら更新の履歴が特殊なファイルに記録される。更新の履歴を格納する特殊なファイルはジャーナル、ログなどと呼ばれる。

【0004】一方、SQLなどのデータベース処理言語はデータベース処理（データベースからの所要データの検索、データベース中のデータの更新、削除、挿入、及びこれらデータ操作を行なうためのデータの定義等）を行なうことを目的とした言語であり、業務で必要となる通信、画面制御などはプログラミングが不可能である。このため、業務プログラムは、データベース処理の部分をSQLで記述し、その他の部分をCOBOLなどの従来の言語で記述して作成されることが一般的である。

【0005】SQL文を含むプログラムをコンパイルすると、SQL以外の部分は通常のロードモジュールに、また、SQLの部分は、SQL手順にコンパイルされる。ここで、元SQL文のあった部分には、データベースシステムのデータベースシステム中核部を経由してコンパイルされたSQL手順を呼び出すための規定の手続き (図15ではsql exec) が替りにおかれて、またその引数としてデータベースシステムに登録されたSQL手順の識別子が与えられる。システムの実現方式にも依存するが、SQLを用いて作成されたロードモジュールとSQL手順は、SQLシステムの定義情報の一部として登録されることが一般的である。

【0006】図1は業務プログラムとデータベースの基本構成を示す図であり、図においては、COBOL業務プログラムの中にSQLプログラムを作成した例を示している。このSQLプログラムはコンパイラによりコンパイルされる。コンパイラがSQLプログラムをコンパイルする場合には、SQLプログラムそれぞれに対して識別子を与える。この図においては、それぞれのSQLプログラムに対応するSQL手順に対して、No. 23とNo. 24が識別子として与えられた例を示している。COBOL業務プログラムは、コンパイルされるとロードモジュールという実行形式の形に変換されるが、このロードモジュール内にはSQL手順を実行するためのCALL文がおかれていている。CALL文は、SQL手順の識別子23と24を引き数にすることにより、カタログに登録されたSQL手順をコールする。こうしてSQL手順に記述されたデータベース処理を実行する。

【0007】即ち、図16に示すように業務プログラムを実行すると、通常のプログラムと同様に実行がなされるが、実行がデータベースシステムの処理要求の手続き (sql exec) に達すると、この手続きからデータベースシステム中核部を経由して対応するSQL手順が取り出され、実行される。このようにして、業務プログラムが適宜データベースにアクセスしながら所要の業務を実現する。

【0008】次に、データベースの定義情報をについて説明する。データベースシステムは、データベースに関する定義情報を種々の方法で生成、管理している。例えばSQLでは、データベース情報をシステムで一元管理することにより、データ定義の重複の防止やデータベース

内部での記憶情報の管理の効率化を実現している。これら定義情報の集まりをカタログと呼ぶ。SQLでは、これらシステムで管理される対象として以下のようなものがある。

(1) 実表。これは従来のファイルに相当するデータの本体である。実表を構成するフィールドを列と呼ぶ。列には各々データ型、長さなどの属性がある。

(2) ビュー。これは、従来の論理ファイルなどに相当し、実表に条件を付して検索した結果をあたかも実表のようにして見せるものである。即ち、データを、あるがままの状態以外に、一次加工、二次加工した形で見せることにより、データへの不要なアクセスの防止、共通したアクセスの一元化などを実現する。これは、複数の実表（ファイル）を結合して一つに見せる、といった複雑な場合も含まれる。また、（複数の）別のビューを参照することにより作成されたビューも存在する。実表とビューを総称して表と呼ぶ。

(3) スキーマ。これは従来のディレクトリなどに相当し、表等を区分けして整理するための入れ物を指す。表（実表とビュー）は、例えばスキーマ名をS1とし、表名をT1とするとS1.T1のようにスキーマ名を冠して名前付けされる。

(4) データベースの利用者。

(5) データベースの利用者が持っている権限。実表やビューに対する操作に対しては、これらを作成したユーザがすべての権限（選択、更新、削除、追加）を有するが、これら権限を適宜他のユーザに与えることによって一定のユーザの集団の中でのデータベースへのアクセスを制御することができる。即ち、SQLではデータのアクセス権限を厳密に管理し、データの参照、更新、挿入、削除などの権限を実表、ビュー毎に、ユーザ毎に制御可能である。また、これらは権限は基本的に実表、ビューなどの生成者が保有するものであるが、これを権限を他人に付与したり、更にこの付与行為までを他人に認めることが可能である。付与行為までを認めるこをWITH GRANT OPTION付きで権限を付与する、と呼ぶ。

(6) データベースの業務プログラムとその中で使用されるSQL手順。

【0009】SQLでは以上の（1）～（6）を例えば以下のような情報としてカタログ中で管理する。

a) ユーザー一覧情報（ユーザ名、保有スキーマ数）

ユーザ名とそのユーザが保有するスキーマの数。

b) スキーマ一覧情報（スキーマ名、所有者名、格納表数）

スキーマの名前、スキーマを所有する所有者の名前、スキーマ中の表の数。

c) 表一覧情報（名前、列数、所属スキーマ名、ビューか実表か）

実表とビューを表として統一的に管理する。表の名前、

表の列の数、表の所属するスキーマ名、実表かビューかの区別からなる。

d) 列一覧情報（名前、所属表名、所属スキーマ名、データ型、長さ）

列毎の情報。列の名前、その列が属する表の名前とその表が属するスキーマの名前、データ型、長さなど。

e) ビュー定義一覧情報（ビュー名、所属スキーマ名、定義本体）

各ビューに対して、その定義本体を管理する。ビューの名前とその属するスキーマ名、及びビュー定義の本体からなる。

f) ビュー対表の参照関係情報（ビュー名、所属スキーマ名、参照表名、参照表所属スキーマ名）

ビューは別の実表またはビューを参照することにより生成される。そのビューが参照しているこれら表毎にビュー名とそのビューが所属するスキーマ名、及びビューが参照している表の名前とその表が所属するスキーマ名がある。

g) 権限一覧情報（付与者、受領者、種類、対象、対象のスキーマ名、再付与可能性）

データベースの使用者に与えられている表に対するアクセスの権限（従来の、読とり可能、書き込み可能、実行可能、などの権限に相当）に関する情報からなる。例えば、権限を付与した人間、権限を受領した人間、権限の種類（当該表に対する選択権、更新権、挿入権、削除権）、権限の対象となる表、権限の付与行為まで与えているかどうか、からなる。特に、新たに表を作成した場合は、作成者はシステムから権限を受領したと見なされ、権限付与者には特別な値（SYSTEM）が設定される。また、ビューを生成する際には、選択権が少なくとも必要である。また、生成されたビューが一定の条件を満たさない（例えば複数の表を参照している、など）場合には、更新が不可能となるため、ビューに対する更新、削除、挿入の権限は生成されない。

h) プログラム一覧情報（業務プログラム識別子、OS内プログラム所在）

業務プログラムの一覧。識別子とプログラムが実際に存在するOS下の位置（ディレクトリ等）からなる。

i) 手順一覧情報（業務プログラム識別子、手順識別子、手順本体）

上記の業務プログラム中で用いているSQLコマンドに對応してコンパイルにより生成されたSQL手順の一覧。プログラム識別子、SQL手順の識別子、及び手順の本体の内容からなる。

j) 手順対表参照関係情報（業務プログラム識別子、手順識別子、スキーマ名、表名）

業務プログラムの各手順が、各々参照している表を管理する。手順とそれが参照する表毎に一つの情報が存在する。

【0010】次に、データベースとカタログの具体例に

ついて説明する。例えば、図17のようなデータベースの内容は図18～図27のようなカタログ情報として管理される。図17に示したデータベースにおいては、鈴木と田中という2人のユーザが存在しており（図18参照）、ユーザ鈴木はスキーマS1を所有しており、ユーザ田中はスキーマS2、S3を所有している（図19参照）。また、スキーマS1は実表T1とビューV1を有しており、スキーマS2は実表T2とビューV2を有しており、スキーマS3はビューV3を有している（図20参照）。また、ビューV1はT1を参照しているビューである（図23参照）。その定義は、

```
SELECT * FROM T1 WHERE t1  
1 > 100
```

であり（図22参照）、更新が可能なビューである（図24参照）。ビューV2はT1とT2を参照しているビューである（図23参照）。その定義は、

```
SELECT t11, t12, t21 FROM T1, T2 WHERE t11 = t23
```

であり（図22参照）、複数の表を参照しているため、更新が不可能なビューである（図24参照）。また、このビューは他人（鈴木）が所有する表T1を参照しているため、予め鈴木から田中に関する選択権の付与がなければならない（図24参照）。ビューV3はV2を参照しているビューである（図23参照）。その定義は、

```
SELECT v21 FROM V2
```

であり（図22参照）、複数の表を参照しているため、更新が不可能なビューである（図24参照）。プログラムPROG1にはSQL文が二つあり、各々P1、P2なるSQL手順である（図25及び図26参照）。プログラムPROG2にはSQL文が一つあり、P3なるSQL手順である（図25及び図26参照）。SQL手順P1はV2を参照している（図27参照）。SQL手順P2はT2を参照している（図27参照）。SQL手順P3はV1とV3を参照している（図27参照）。鈴木は田中にT1に関する選択権（参照する権利）を再付与権つきで与えている（図24参照）。以上のように、図18から図27に示したカタログ情報は、前述したデータベースの定義情報の具体例を示すものである。

【0011】更に、SQLなどを用いたデータベースシステムは、以下のような特長を持つ。

（1）データベースに対して実行された更新、削除などの履歴をすべてジャーナルに記憶することにより、万一のデータベースの破壊に対してもデータベースを必ず回復できる。

（2）更に、索引の追加、アクセス権限の変更など、データベースに変化が生じ、カタログ中に登録されているSQL手順が最適な手続きでなくなったり場合には、SQL手順がCOBOL業務プログラムを通じて実行される際に当該SQL手順が動的に再コンパイルされ、変化に対応した最適な手続きを再度生成し、これをカタログに

登録しなおして実行を再開する。

【0012】

【発明が解決しようとする課題】以上の特長を実現する一方で、逆に以下のような問題点が生じている。

（1）あるシステムの上で開発した業務をパッケージ化しようとすると、他の開発中の業務などを含め、すべてのデータベースの定義情報を一つのカタログ内で一元管理されているため、その中の必要な部分だけをとりだしてパッケージにまとめることが困難である。例えば、実表は複数のビューに参照されており、またビューは複数のビューに参照されている。また、ビューが参照するに際しては権限が必要であり、これら権限が定義されている状態をビューを生成するたびに忠実に再現しなければならない。例えば、上記の例では、ビューV2を生成するためには、表T1、表T2の作成が必要であり、更にこれら表の作成には、前もってスキーマS1、S2と各々のスキーマの所有者である鈴木、田中のユーザが生成されなければならない。また、これら表が生成された後、更に表T1に関する選択権を鈴木から田中に対して付与しておかなければ、田中はT1を参照することができず、ビューV2が生成できない。このように、定義情報は相互に複雑にからみあっており、また時間軸に沿って履歴的な復元が必要となる。現在は、

a) これを人間が必要な部分、必要な順序を手作業で認識し、取り出して、

b) 更に、手作業により移行先で注意深く再現を行なっている。即ち、従来より行なわれているソフトウェアのパッケージの作成、普及が困難であり、対象システム各々において人手による複雑なシステムインストール、開発作業が必要となっていた。

【0013】（2）更に、最近は、LAN（ローカルエリアネットワーク）を通じて多くのマシンが接続され、ソフトウェア開発もあるマシンで行なわれるとその成果がLANを通じて複数の別のマシンに配布される、といった形態が普及している。しかし、上記のような問題があり、データベースをアクセスする業務プログラムはこのような仕組みが整ってもこれを有効に活用して、ソフトウェア配布を行なうことが困難であった。

【0014】（3）更に、最近は、ソフトウェア開発全体を管理するCASE（Computer Aided Software Engineering）が普及しつつあるが、CASEの中で開発作業の定義情報を管理するリポジトリは上記のカタログと同様な位置付けのものであり、これら開発作業環境の移行や複数グループによる開発の分散を行う場合にも上記（1）の問題が生じている。

【0015】（4）これらの定義情報を履歴に沿って移行先で再現するためにはデータベースシステム自身に従来の機能に加えて特殊な機能が必要になる可能性がある。例えば、各スキーマ中に表がいくつある、といった

情報を移行しようとすると、履歴にカタログの内容をそっくり保管し、パッケージインストール時にこの部分を取り出し、移行先で、システムのカタログの該当部分を直接変更することが可能である。しかし、このためには、通常のデータベース機能以外のカタログ処理の特殊なインターフェースを持つ必要がある。また、このため、COBOL、SQLと行った標準言語で開発されたパッケージであっても、特殊なインターフェースがベンダの仕様に依存してしまい、異なるベンダ間ではシステムの移行ができない。

【0016】(5) また、従来のジャーナルはデータベース全体を通じてのデータ変更の履歴を記録しており、データベースの保管の際には、その復元作業のため常にデータベース全体を保管（バックアップ）する必要がある。データベース全体の内のごく一部の部分にのみ変更が加えられていたと判っていても、常に全体を保管するため、システムの保守運用時間が長くなっている。

【0017】(6) また、システムの移行に際しては、業務プログラムとカタログ中に登録されたSQL手順との対応関係を維持するための識別子の生成が困難となる。特に、業務プログラムを別のマシン、別のSQLシステムに移行した場合には、そのマシン上で既に稼働中の業務プログラム・SQL手順が使用している識別子と、移行により持ち込んだ識別子が衝突する場合があり、移行が困難である。また、移行先では移行したSQL手順がデータ量などの関係で必ずしも最適なSQL手順にコンパイルされた状態ではない場合がある。

【0018】この発明は以上のような問題点を解決するためになされたもので、データベースシステムのプログラム及びデータの退避、移行、復元が容易に行える装置を得ることを目的とする。

【0019】

【課題を解決するための手段】第1の発明に係るデータベースシステム退避装置は、退避命令とそのパラメータを入力するコマンド解釈手段と、コマンド解釈手段により解釈された退避命令とそのパラメータに基づいて退避すべき対象物を特定する制御手段と、この制御手段により特定された退避すべき対象物の退避手順を生成する退避手順生成手段を有したものであり、以下の要素を有するものである。

(a) 上記データベースシステムをアクセスするデータベースアクセス手段、(b) 退避すべきデータを保存データとして保存媒体に保存する保存媒体アクセス手段、(c) 上記データベースシステムの退避命令を入力して解釈するコマンド解釈手段、(d) 上記データベースアクセス手段によりデータベースシステムの定義情報をアクセスすることにより、退避すべき対象物の退避すべき手順を生成する退避手順生成手段、(e) 上記コマンド解釈手段により解釈された退避命令に基づき、上記データベースシステムの中から退避すべき対象物を特定する

とともに、上記退避手順生成手段を動作させ、退避手順生成手段により生成された退避すべき手順に従って、退避すべき対象物を上記データベースアクセス手段によりデータベースシステムから取り出し、上記保存媒体アクセス手段により保存媒体に保存する制御手段。

【0020】第2の発明に係るデータベースシステム退避装置は、退避命令のパラメータとして、退避すべきユーザ名、スキーマ名、表名、ビュー名、プログラム名、権限名のいずれかを入力すると共に、これら退避すべき対象物の種類の限定と範囲の指定がなされることを特徴とし、制御手段はこの対象物の指定と、この対象物に対する種類と範囲の指定に基づいて、定義情報から対象物を特定することを特徴とするものである。

【0021】第3の発明に係るデータベースシステム退避装置は、データベースシステムがSQL等のデータベース処理言語を用いてアクセスでき、データベースアクセス手段はSQL等のデータベース処理言語を用いて、データベースシステムをアクセスすることを特徴とするものである。

【0022】第4の発明に係るデータベースシステム退避装置は、保存媒体に情報を保存する場合に、この情報をSQL等のデータベース処理言語により記述して保存することを特徴とするものである。

【0023】第5の発明に係るデータベースシステム退避装置は、定義情報を退避する場合にSQL等のデータベース処理言語により、定義情報を生成するための定義コマンドに変換して退避することを特徴とするものである。

【0024】第6の発明に係るデータベースシステム退避装置は、データを退避する場合にデータをSQL等のデータベース処理言語によるデータ挿入のためのデータ挿入コマンドに変換して退避することを特徴とするものである。

【0025】第7の発明に係るデータベースシステム退避装置は、ユーザ名やファイル名等の置換可能な変数が退避すべき対象物に存在する場合に、これらユーザ名やファイル名等の置換可能な変数を置換情報として生成しこれを保存することを特徴とするものである。

【0026】第8の発明に係るデータベースシステム退避装置は、データを処理するプログラムはSQL等のデータベース処理言語による処理手順を呼び出して実行すると共にそのプログラムが他の機種に移植された場合のことを考慮して、予めデータベース処理言語による処理手順をソースコードとしてプログラム内に保持していることを特徴としており、定義情報内にある実行形式の処理手順が退避すべき対象物になった場合でも、この実行形式の処理手順を退避せずに、その処理手順のソースコードを有するプログラムを退避することを特徴とするものである。

【0027】第9の発明に係るデータベースシステム退

避装置は、データベースのデータを退避せず単にデータベースのデータ記憶領域を確保するための領域確保コマンドを生成して保存することを特徴とするものである。

【0028】第10の発明に係るデータベースシステム退避装置は、データベースシステムがアクセス履歴情報を記録するジャーナルを有しており、アクセス履歴情報は、例えば表名等の識別子と共に記録されており、制御手段は退避すべき対象物に関連したアクセス履歴情報のみを表名などの識別子を元に検索し退避するようにしたものである。

【0029】第11の発明に係るデータベースシステム退避装置は、ユーザ、スキーマ、表、ビュー、権限、プログラム間の依存関係を特定する退避手順生成手段を有し、この退避手順生成手段が生成した依存関係を依存情報として保存する制御手段を有することを特徴とするものである。

【0030】第12の発明に係るデータベースシステム復元装置は、前述したデータベースシステム退避装置により退避された保存データを入力することにより、データベースシステムの一部または全部を復元する事を特徴とするものである。

【0031】第13の発明に係るデータベースシステム移行装置は、前述したようなデータベースシステム退避装置により退避された保存データを入力することにより、他の機種或いは他のマシン上に新たにデータベースシステムを構築することを特徴とするものである。

【0032】第14の発明に係るデータベースシステム復元/移行装置は、前述したデータベースシステム退避装置がSQL等のデータベース処理言語により記述されたコマンドの形式でデータを退避している場合に、このデータベース処理言語により記述されたコマンドを入力して実行することにより、データベースシステムを復元または移行することを特徴とするものである。

【0033】第15の発明に係るデータベースシステム移行装置は、前述したデータベースシステム退避装置において、保存したデータの中に置換可能な変数が存在することを示す置換情報が存在する場合に、その置換情報に基づき置換可能な変数に新たな変数を与えて、変数の値を置き換えながらデータベースシステムを新たに構築することを特徴とするものである。

【0034】第16の発明に係るデータベースシステム移行装置は、前述したデータベースシステム退避装置が実行形式の処理手順を保存せず、かわりに処理手順のソースコードを有しているプログラムを保存している場合に、実行時に処理手順が定義情報に存在しないことを条件として、実行時に処理手順のソースコードをコンパイルして実行形式の処理手順を新たに定義情報に作成することを特徴とするものである。

【0035】第17の発明に係るデータベースシステム復元/移行装置は、前述したデータベースシステム退避

装置により退避された保存データを入力するとともに、復元仕様又は移行仕様を入力し、上記保存データの依存関係を参照することにより、復元又は移行するデータを復元仕様又は移行仕様に適合するように評価して、データベースを復元又は移行する復元/移行手順生成手段を有することを特徴とするものである。

【0036】

【作用】第1の発明におけるデータベースシステム退避装置においては、制御手段が退避命令に指定されたパラメータから保存すべき対象物を特定すると共に、退避手順生成手段が定義情報を参照することにより、退避すべき手順を時間軸に沿って矛盾なく生成する。従って他の業務開発中、あるいは他の業務が稼働中であるデータベースシステムから、指定した業務に必要な定義情報を時間軸、相互の関係を保存したまま抽出できる。これを用いて業務動作に必要な情報を復元、移行することができる。

【0037】第2の発明におけるデータベースシステム退避装置は、退避命令としてユーザ名、スキーマ名、表名、ビューネーム、プログラム名、権限名のいずれか1つが指定されると共にその種類と範囲が指定され、制御手段は退避命令による対象物と、その種類と範囲に基づいて、退避に必要な対象物を定義情報から判定して、退避すべき対象物として特定するので、コマンド解釈手段に入力する退避命令には必要最少限の情報を入力することにより、自動的に退避すべき対象物が特定されることが可能になる。

【0038】第3の発明に係るデータベースシステム退避装置は、データベースシステムがデータベース処理言語によりアクセスでき、データベースアクセス手段がデータベース処理言語を用いてデータベースシステムをアクセスするため、特別なデータベースアクセス手段を持つ必要が無く、従来のデータベースシステム及び従来のデータベースアクセス手段をそのまま用いてデータベースシステムを退避することが可能になる。

【0039】第4の発明におけるデータベースシステム退避装置は、データベースシステムの情報を退避する場合に、SQL等のデータベース処理言語によるコマンドに変換して保存するため、これを入力してデータベースを復元する場合、或いは別なシステムにデータベースシステムを移行する場合に、特別な復元手段或いは、移行手段というものを開発する必要が無く、従来のデータベース処理言語を処理できる手段が備わっていれば、データの復元或いはデータの移行が可能になる。

【0040】第5の発明におけるデータベースシステム退避装置は、定義情報を退避する場合に、SQL等のデータベース処理言語により定義情報を生成するための定義コマンドに変換して保存するため、これを入力してデータベースを復元する場合、或いは別なシステムにデータベースシステムを移行する場合に、特別な復元手段或

いは、移行手段というものを開発する必要が無く、従来のデータベース処理言語を処理できる手段が備わっていれば、データの復元或いはデータの移行が可能になる。

【0041】第6の発明におけるデータベースシステム退避装置は、第5の発明におけるデータベース退避装置が定義情報に対してデータベース処理言語による定義コマンドを用いたのに対して、データを退避する場合にはデータベース処理言語によるデータ挿入のためのデータ挿入コマンドに変換して退避するものである。従って、データを復元する場合、或いはデータを新たなシステムに移行する場合でも、従来からのデータベース処理言語を処理できる手段が備わっていれば、データを復元或いは移行することが可能になる。

【0042】第7の発明におけるデータベースシステム退避装置は、データベースシステムを他のシステムに移行するような場合に、移行先の他のシステムに既に同一ユーザ名や同一ファイル名等が存在する場合に、これらの名が重複することを避けることを可能にしたものであり、ユーザ名やファイル名等の置換可能な変数を予め置換情報として生成し、保存するものである。この置換情報を用いることにより、保存した情報を新たなシステムに移行する場合に、変数の値を、移行しようとして居るシステムに既に存在しているユーザ名やファイル名と重複しない名前に変えることが可能になる。

【0043】第8の発明におけるデータベースシステム退避装置は、定義情報にある実行形式の処理手順を退避することなく、またデータベースシステムにアクセスするプログラムに処理手順のソースコードをもたせておき、このプログラムを退避することを特徴としている。このように実行形式の処理手順を退避しないことにより、退避するデータ量を削減することが可能になる。また処理手順をソースコードで退避することにより、処理手順移行先のシステムの環境で再コンパイルすることが可能になり、移行先のシステム環境に最適な実行形式の処理手順を生成することができる。

【0044】第9の発明におけるデータベースシステム退避装置は、データベースのデータを退避することなく、データベースのデータ記憶領域のみを確保するための領域確保コマンドを生成するため、退避すべきデータ量の削減が図れると共に、他のシステムにデータベースを移行する場合、定義情報が移行されると共に、データベースの領域のみが確保されるため、移行先においてデータを新規に蓄積していくような場合に、移行元のデータを移行せず領域のみを確保することが可能になる。

【0045】第10の発明におけるデータベースシステム退避装置は、従来のジャーナルを退避する場合にアクセス履歴情報をすべて退避していたのに対して、識別子により関連するアクセス履歴情報を選択できるようになる。従って制御手段が特定した退避すべき対象物に関連するアクセス履歴情報のみが退避可能になる。

【0046】第11の発明におけるデータベースシステム退避装置は退避手順生成手段がユーザ、スキーマ、表、ビュー、権限、プログラム間の依存関係を特定し、制御手段がこの依存関係を保存するため、この保存されたデータからデータを復元する場合、或いは移行する場合に依存情報を参照することにより、データの選択、或いは削除が可能になる。即ち、依存関係とは、データベースシステムの定義情報に示されている各データベースの構成要素の関連を抽出したものであり、この依存関係を参照するということは、即ちデータベースシステムの定義情報を参照するということに等しく、この依存関係を保存データ内に保存しておき、復元移行の際に、この依存関係を参照するということはデータベースシステムの定義情報を参照するに等しく、結果として、保存されたデータの中から更に別なデータを抽出したり、或いは余分なデータを削除する場合に矛盾なく抽出或いは削除することが可能になる。

【0047】第12の発明におけるデータベースシステム復元装置は、データベースシステム退避装置により保存されたデータを入力して、データベースシステムを復元するものであり、データベースシステム退避装置が保存媒体に対して、データを時間軸或いは相互の依存関係を矛盾なく保存しているため、この保存データを順に入力することにより、元のデータベースシステムを復元することが可能になる。

【0048】第13の発明におけるデータベースシステム移行装置は、同様に保存媒体が時間軸或いは相互関係において、データを矛盾無く保存しているために、移行の際には保存されたデータを順に新たなシステムにインストールしていくことにより、必要な定義情報、或いは必要なデータを新たなシステム先に構築することが可能になる。

【0049】第14の発明におけるデータベースシステム復元/移行装置は、保存されたデータがSQL等のデータベース処理言語によるコマンドによって記述されているため、このコマンドを実行することにより、定義情報或いはデータベースのデータ等を復元または移行することが可能になる。このようにデータベース処理言語により記述されたコマンドを実行するだけで、定義情報、或いはデータ等を復元、または移行することが可能ため、従来から存在するデータベース処理言語の処理手段をそのまま用いることにより、データベースシステムの復元移行が可能になる。従ってデータベースシステムを復元または移行するための特別な復元手段、或いは特別な移行手段というものが不要になる。

【0050】第15の発明におけるデータベースシステム移行装置は、データベースシステムを他のシステムに移行する場合であって、移行先のシステムに既に重複するユーザ名やファイル名等が存在する場合に、置換情報にある変数の値を置き換えることが可能になるため、移

行先においてユーザ名やファイル名等が重複することなく、システムの移行をすることが可能になる。

【0051】第16の発明におけるデータベースシステム移行装置は、データベースシステムを新たなシステムに移行する場合に、実行形式の処理手順を保存データから移行するのではなく、保存されたプログラム内にあるソースコードをコンパイルすることにより、実行形式の処理手順を定義情報内に作成する。従って、移行先のシステム環境にあった最適な実行形式の処理手順が生成される。

【0052】第17の発明におけるデータベースシステム復元／移行装置は、復元仕様又は移行仕様を入力し、復元／移行手順生成手段が上記保存データの依存関係を参照することにより、復元又は移行するデータを復元仕様又は移行仕様に適合するように評価して、データベースを復元又は移行するので、保存データすべてを用いてシステムの復元、移行をするのではなく、保存データの一部を用いてシステムの復元、移行をすることが可能になる。

### 【0053】

#### 【実施例】

実施例1. 従来例で説明したように、データベースの処理言語として、SQLと呼ばれる国際標準言語が用いられることが一般的である。このため、以下では関係データベースと国際標準データベース処理言語SQLを前提として説明するが、内容的には他のデータベース、あるいは他のデータベース言語でも同様に適用できる。又、この実施例では、従来例で示した図17～図27に示したデータベースシステムを具体例として説明する。図1はこの発明の一実施例を示すデータベース及びプログラム退避／移行／復元装置の一例を示す図である。ここで退避とは、既に存在しているデータベース及びプログラムの一部又は全部を磁気テープや磁気ディスク等の保存媒体に複写することをいう。又移行とは、前述した退避により保存媒体に複写してあるデータベース及びプログラムを用いて、ある計算機システムにおいて動作しているデータベースシステムを他の計算機システムに移植することを言う。又復元とは、前述した退避により保存媒体に複写してあるデータベース及びプログラムを元の計算機システムに複写し直すことをいう。ここで、移行と復元の違いは、移行が他の計算機システムに対して行われるのに対して、復元が元の計算機システムに対して行われる点である。

【0054】以下、図に基づいてデータベース及びプログラム退避／移行／復元装置の各部について説明する。

#### 1) コマンド解釈装置

ユーザへのインターフェースを提供し、ユーザからの退避、復元、移行などの要求とその対象物を受取り、正当性を確認してこれら要求を制御部に送る。又、移行の場合は、必要に応じてユーザ名などの置換指示をユーザよ

り受取り、これを制御部に送る。

#### 【0055】2) 制御部

装置全体を制御すると共に、データベース及びカタログアクセス装置を用いて、ユーザから指定された退避、復元、移行の対象物を特定する。

【0056】退避の場合には、退避移行復元手順生成装置を呼び出すことによりこれら特定された対象物と処理要求から必要な定義情報、置換情報、依存情報を生成してこれをSQLの実行形式にして保存媒体アクセス装置により保存媒体に順次記憶する。更に、指定によりデータ自身（実表の内容）をデータベース及びカタログアクセス装置により読みだし、これをSQLの実行形式に変換して保存媒体アクセス装置により保存媒体に記憶する。更に、指定によりジャーナルをデータベース及びカタログアクセス装置により読みだし、これを保存媒体アクセス装置により保存媒体に記憶する。更に、指定によりプログラムをプログラムアクセス装置により読みだし、これを保存媒体アクセス装置により保存媒体に記憶する。また、以上の動作の際、ビューを評価して実表とする指定がある場合、あるいはデータベースの実表の内容に対してその一部をSQLコマンドの実行により変形したい要求がある場合には、適宜ビューSQL実行装置によりこれを実行する。

【0057】復元移行の場合には、保存媒体から定義情報、置換情報、依存情報を読みだし、これら情報と指定された対象物から退避移行復元手順生成装置を呼び出すことにより復元移行対象を選択する。これら特定された対象物に対して置換が必要な場合、置換する内容をコマンド解釈装置を経由してユーザから入力する。次に指定により置換情報に従って必要な置換を行いながら、特定された対象物に関する定義情報を保存媒体から保存媒体アクセス装置により読みだし、データベース及びカタログアクセス装置により復元移行する。これら定義情報はSQLの定義系のコマンドの形となっているため、この操作はこれらコマンドを実行することによりなされる。更に、指定によりデータ自身（実表の内容）及びジャーナルを保存媒体アクセス装置により保存媒体から読みだし、これをデータベース及びカタログアクセス装置によりシステムに復元移行する。更に、指定によりプログラムを保存媒体アクセス装置により保存媒体から読みだし、これをプログラムアクセス装置によりデータベースシステムに記憶する。また、以上の動作の際、ビューを評価し、実表とする指定がある場合、あるいはデータベースの実表の内容に対してその一部をSQLコマンドの実行により変形したい要求がある場合には、適宜ビューSQL実行装置によりこれを実行する。

#### 3) 退避移行復元手順生成装置

制御部から対象物を受取り、これらに対して必要な定義情報とその順序を生成する。退避の場合には、これら生成に必要な情報はデータベース及びカタログアクセス装

置からカタログをアクセスすることにより得られる。復元移行の場合には、これら生成に必要な情報は保存媒体中にある依存情報をアクセスすることにより得られる。

#### 4) ビューSQL文実行装置

指定されたビュー、またはSQL文を実行し、その結果を制御部に送る。

#### 5) 保存媒体アクセス装置

保存媒体（磁気テープ、ディスク等）に対して指定した内容を読みだし、書き込みする装置。

#### 6) データベース及びカタログアクセス装置

データベースの内容及びカタログの内容を読みだし、書き込みする装置。この場合には、データベースシステムに対して必要なSQL文を実行することで機能が実現される。また、この装置によりジャーナルの内容の読みだし、またはジャーナルの内容をデータベースに適用し、

データベースを最新の状態にする。これらジャーナル操作に対するSQLの規格は存在しないが、システム毎に何等かの手段が用意されている。例えば、ジャーナルの読みだし、設定は通常のOSのファイル読みだし、生成により実行し、ジャーナルの実行はRECOVERコマンドなどによりデータベースシステムが実行する。

#### 7) プログラムアクセス装置

業務プログラムを読みだし、設置する装置。通常のOSのファイルのアクセス機能で充分である。

#### 【0058】次に動作について説明する。

##### 1) コマンド解釈装置の動作

ユーザからの退避移行復元要求はまずコマンド解釈装置により解釈され、内容の正当性が確認されて、これらを内部形式に変換し、制御部に送る。コマンド解釈装置は、以下のような指定を受け付ける。

動作指定	=BACKUP：退避動作 RESTORE：復元、移行先インストール動作
対象物のリスト	=対象物、対象物、...
対象物	=ユーザ名（ユーザの所有物全ての指定の省略形と解釈） スキーマ名（スキーマ下の全ての表を指定したと解釈） スキーマ名、表名（表を指定したと解釈）
権限名	権限名
対象種類指定	プログラム名 =D：データ C：カタログ P：プログラム これらの組み合わせ、例えばD+P等
対象範囲指定	=ONLY：指定された対象物のみを対象物とする。 DOWN：指定された対象物とこれらが参照している対象物すべてを対象物とする。 ALL：指定された対象物とこれらが参照している対象物すべてとこれらを参照している対象物すべてを対象物とする。
制限範囲指定	=ユーザ名リスト（これらユーザ以外のユーザのスキーマ、表等を対象としない） スキーマ名リスト（これらスキーマ以外のスキーマ、表等を対象としない）

ビューの内、それを実行して実表としたいもののリスト。実表の内、それに対してデータの選択、変形を行うためのSQL文のリスト。

#### 【0059】2) 制御部の動作

制御部は、コマンド解釈装置からの要求を解釈し、退避、移行、復元操作の全体を制御する。

##### 【退避の場合】

(1) 制御部は、コマンド解釈装置から受け付けた対象物を退避移行復元手順生成部におくり、対象物の一覧と定義情報、置換情報、依存情報を受け取る。対象物の一覧として以下のものがある。

U：ユーザー一覧  
S：スキーマ一覧  
T：実表一覧

V：ビュー一覧  
A：権限一覧  
P：プログラム一覧  
RU：制限ユーザー一覧  
RS：制限スキーマ一覧

(2) ビュー一覧V中のビューで、ビューを評価し、実表とする指定のあるものは、これを評価し、このビューを実表とする。定義情報、依存情報において、このビューに対応する情報を実表の情報に置き換える(CREATE VIEW文をCREATE TABLE文に置き換える)。また、この実表を記憶しておく。また、このビューが参照しているビュー及び実表のうち、このビューのみにより参照されているものを定義情報から除く。

(3) 対象種類指定にC（カタログ）があれば、置換情

報、定義情報、依存情報を保存媒体アクセス装置により保存媒体に格納する。

(4) 対象種類指定にD(データ)があれば、実表一覧Tにある実表の内容をデータベース及びカタログアクセス装置によりとりだし、これをINSERTコマンドとして保存媒体アクセス装置により保存媒体に記憶する。保存媒体に記憶する実表がビュー評価により(2)で生成されたものであれば、(2)で記憶しておいた実表の内容を保存媒体に記憶する。また、保存媒体に記憶する実表に対してSQL文の実行の指定があればこれをビューSQL文実行装置により実行し、その結果をINSERTコマンドの形にして保存媒体に記憶する。また、対象種類指定にD(データ)があれば、ジャーナルをデータベース及びカタログアクセス装置によりとりだし、これを保存媒体アクセス装置により保存媒体に記憶する。この際、ジャーナルには操作対象となった実表の識別子が付されており、実表一覧Tにある表に関するジャーナル内容のみを取り出す。対象種類指定にP(プログラム)があれば、P(プログラム)にあるプログラムをプログラムアクセス装置によりとりだし、これを保存媒体アクセス装置により保存媒体に記憶する。

#### 【0060】[復元移行の場合]

(1) 保存媒体アクセス装置から定義情報、置換情報、依存情報を読み出す。

(2) 制御部は、コマンド解釈装置から受け付けた対象物と依存情報を退避移行復元手順生成部におくり、対象物の一覧を受け取る。対象物の一覧は、

U: ユーザー一覧  
S: スキーマ一覧  
T: 実表一覧  
V: ビュー一覧  
A: 権限一覧  
P: プログラム一覧

RU: 制限ユーザー一覧

RS: 制限スキーマ一覧

(3) コマンド解釈装置により置換指定をユーザから受け取り、その置換内容で置換ファイルを置き換える。

(4) 対象種類指定にC(カタログ)があれば、置換情報により定義情報を置き換えながら、対象物に関する定義情報をデータベース及びカタログアクセス装置により復元、移行する。この場合は、SQLの定義コマンドCREATE USER文群、CREATE SCHEMA文群、CREATE TABLE、CREATE VIEW、GRANT文群を読みだし、置換情報の内容に従って置き換えを行ないながら、これら文を記憶されている順に実行する。これら文は既に退避の際に時間的、空間的に矛盾のない順に記憶されているため、この順に実行を行なうだけでよい。ビュー一覧V中のビューで、ビューを評価し、実表とする指定のあるものは、CREATE VIEWコマンドからCREATE TABLEに変換してこれを実行する。また、これをSQL文の一つとして記憶しておき、後に(5)で実行する。また、このビューが参照しているビュー及び実表のうち、このビューのみで参照されているものを定義情報から除く。これは依存情報中で、このビューのみに依存しているものをたどることにより識別できる。

【0061】(5) 対象種類指定にD(データ)があれば、実表一覧Tにある実表の内容であるINSERT文をデータベース及びカタログアクセス装置によりとりだし、これを実行してデータベースに復元する。また、この実表に対してSQL文の実行の指定があればこれをビューSQL文実行装置により実行し、その結果をT'として保存し、この表の内容を、

DELETE \* FROM 表名  
で一旦廃棄し、

#### INSERT INTO

により先の結果で置き換える。この具体例について以下に説明する。例えば実表T1が存在し、これを参照するビューV1が存在するものとする。即ち、実表T1に対してビューV1がSQL文の実行により評価される。もし、実表T1がビューV1のみにより参照され、それ以外のビューには一切参照されないことが依存情報等を参照することにより判明した場合には、実表T1をデータベースに記憶しておく必要はない。従って、その実表T1を削除しても構わなくなる。この場合には、ビューV1をT1に対して評価した結果をT'として一時記憶しておき、実表T1が他のビューや他のプログラムにおいて参照されることがないと判断された時点で実表T1を削除し、T'に記憶しておいたビューV1を実表として生成することが可能である。また、実表T1のデータの内、例えばt11=100のデータのみを復元する場合などは、

表名

SELECT \* FROM T'  
SELECT \* FROM T1

WHERE t11=100

を直接実行し、その結果でT1の元のデータを置きかえるものである。

(6) 対象種類指定にD(データ)があれば、保存媒体からジャーナルをとりだし、これをデータベースに適用する。この際、ジャーナルには操作対象となった実表の識別子が付されており、実表一覧Tにある表に関するジャーナル内容のみを取り出す。

(7) 対象種類指定にP(プログラム)があれば、P(プログラム)にあるプログラムをプログラムアクセス装置によりとりだし、これを保存媒体アクセス装置により保存媒体に記憶する。

【0062】尚、データベースをあるマシンから別マシンへ移行する場合はこれら全ステップを実行する。またデータベースを同一マシン上で復元するためには

(5)、(6)の二つのステップを実行すればよい。これらは対象種類の指定でD(データ)のみを指定することにより実現される。また、指定により現在のデータ内容を破壊して復元する場合には、

DELETE \* FROM 表名

によりデータベース内容を一旦破壊して復元を行う。

【0063】データベースの内容自身の移行には、IN SERTコマンドによりSQLコマンドとして運ぶのではなく、システムにより更に高速な形で保存することも可能である。例えば、データの内容をファイルの形で保存し、復元の時もファイルのまま直接置き換えを行う、などが可能である。この場合いは上記の復元操作におけるDELETE文などは不用となる。この後、業務が開始され、復元された業務プログラムを実行する。この際、SQLの手順一覧情報は移行しないため、移行先ではSQL手順にアクセスしようとしてカタログ中にこれが見つからない。このため、システムは当該SQL手順を再度コンパイルし、これを手順一覧情報に登録して実行を再開する。これのように、SQLシステムの持つ動的再コンパイル機構を利用することで、手順一覧情報も自動的に再現できることになる。

【0064】次にコマンド解釈装置が受け付ける指定と、その指定により制御部が退避移行復元手順生成装置から受け取る対象物に関する具体例について説明する。例えば、コマンド解釈装置に受け付けられる指定はいろいろな組み合わせが可能である。例えば、この指定した業務に必要な定義情報の取りだし方として以下のような場合を考えられる。

a) [必要な要素を全て指定する場合]

業務を構成しているユーザ、スキーマ、表、ビュー、プログラム、権限、ユーザを全て指定し、これによりシステムが指定されたこれら情報を、相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

b) [ユーザ名を与えて他はシステムが自動的に抽出する場合]

業務を構成する一つまたは複数のユーザ名のみを与えて、このユーザが所有するスキーマ、表、ビュー、プログラム、権限を自動的に抽出し、これら情報を、相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

c) [スキーマ名を与えて他はシステムが自動的に抽出する場合]

業務を構成する一つまたは複数のスキーマ名のみを与えて、このスキーマを所有するユーザ、及びスキーマの中にある所有する表、ビュー、権限を自動的に抽出し、これら情報を、相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

d) [表名を与えて他はシステムが自動的に抽出する場合]

業務を構成する一つまたは複数の表名のみを与えて、こ

の表が属するスキーマ、及びそのスキーマを所有するユーザ、その表に関する権限を自動的に抽出し、これら情報を、相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

e) [ビュー名を与えて他はシステムが自動的に抽出する場合]

業務を構成する一つまたは複数のビュー名のみを与えて、このビューが属するスキーマ、及びそのスキーマを所有するユーザ、そのビューに関する権限、及びこのビューが参照しているビュー及び表、これら参照に仮定されている権限を自動的に抽出し、これら情報を、相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

f) [プログラム名を与えて他はシステムが自動的に抽出する場合]

業務を構成する一つまたは複数のプログラム名のみを与えて、このプログラムが参照しているユーザ、スキーマ、表、ビュー、権限を自動的に抽出し、これら情報を、相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

g) [権限名を与えて他はシステムが自動的に抽出する場合]

業務を構成する一つまたは複数の権限名のみを与えて、この権限及び権限を付与しているユーザ、権限が付与されているユーザ、権限対象の表、ビューを自動的に抽出し、これら情報を、相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

h) 上記のa)～g)において、更に抽出された表、ビューを参照しているビューの全体をも抽出し、これらビューの生成に必要なスキーマ、ユーザ、権限を抽出して、これらを含めて相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

i) 上記のa)～g)及びそれらとh)の組み合わせにおいて、抽出された情報を参照しているプログラムの全体を抽出する。

【0065】次に従来例で示したデータベースの定義情報の例を用いてコマンド解釈装置に対して指定されたものに対して制御部が定義情報を参照して特定するユーザ一覧U、スキーマ一覧S、実表一覧T、ビュー一覧V、権限一覧A、プログラム一覧Pの値をいくつかの例で示す。以下に述べる例は前述したコマンド解釈装置が受け付ける指定の例として述べた(a)から(i)には直接対応していないが、その中の1つ、或いは複数を組み合わせて指定した場合を示している。また以下の例においては、説明を簡単にするためにコマンド解釈装置が受け付ける全ての指定を示しているものではなく、単に対象物のリスト、対象範囲、制限範囲指定の3つの指定のみを例示している。また、ユーザ一覧U、スキーマ一覧S、ビュー一覧V、実表一覧T、権限一覧A、プログラム一覧P、制限ユーザ一覧RU、制限スキーマ一覧RS

は制御部が退避移行復元手順生成装置から受け取る値をそれぞれ示すものである。退避移行復元手順生成装置は、コマンド解釈装置に入力された指定から、これらユーザー一覧U、スキーマー一覧S、実表一覧T、ビュー一覧V、権限一覧A、プログラム一覧P、制限ユーザー一覧R U、制限スキーマー一覧R Sを求め制御部に出力するが、その手順については後述する。また、例中に示すの記号は全体の意味を示すものであり、特に指定がされないことを示すものである。

### 【0066】

例1 指定した実表を退避

対象物のリスト=鈴木、S 1、

S 1. T 1

対象範囲 =ONLY

制限範囲指定 =なし

ユーザー一覧U	=鈴木
スキーマー一覧S	=S 1
ビュー一覧V	=φ
実表一覧T	=T 1
権限一覧A	=φ
プログラム一覧P	=φ
制限ユーザー一覧R U	=φ
制限スキーマー一覧R S	=φ

### 【0067】

例2 複数の指定した実表を退避

対象物のリスト=鈴木、田中、S 1、S 2

S 1. T 1, S 2. T 2

対象範囲 =ONLY

制限範囲指定 =なし

ユーザー一覧U	=鈴木、田中
スキーマー一覧S	=S 1、S 2
ビュー一覧V	=φ
実表一覧T	=T 1、T 2
権限一覧A	=φ
プログラム一覧P	=φ
制限ユーザー一覧R U	=φ
制限スキーマー一覧R S	=φ

### 例4 複数のビューと参照している実表

対象物のリスト=鈴木、田中、S 1、S 2

S 1. V 1, S 2. V 2,

S 1. T 1, S 2. T 2

対象範囲 =ONLY

制限範囲指定 =なし

ユーザー一覧U	=鈴木、田中
スキーマー一覧S	=S 1、S 2
ビュー一覧V	=V 1、V 2
権限一覧A	=T 1 に関する田中の選択権
実表一覧T	=T、T 2

### 【0068】

例3 ビューとその参照している実表

対象物のリスト=鈴木、S 1

S 1. V 1, S 1. T 1

対象範囲 =ONLY

制限範囲指定 =なし

ユーザー一覧U	=鈴木
スキーマー一覧S	=S 1
ビュー一覧V	=V 1
実表一覧T	=T 1
権限一覧A	=φ
プログラム一覧P	=φ
制限ユーザー一覧R U	=φ
制限スキーマー一覧R S	=φ

### 【0069】

プログラム一覧P =  $\emptyset$   
制限ユーザー一覧R U =  $\emptyset$   
制限スキーマ一覧R S =  $\emptyset$

【0070】例3に対し、ビュー名のみを指定し、ビューが参照している表名を自動的に取り出すことを指定す

るが例5である。

例5 ビューを退避、関連実表も自動退避

対象物のリスト=S 1. V 1  
対象範囲 =DOWN  
制限範囲指定 =なし  
ユーザー一覧U =鈴木  
スキーマ一覧S =S 1  
ビュー一覧V =V 1  
実表一覧T =T 1  
権限一覧A = $\emptyset$   
プログラム一覧P = $\emptyset$   
制限ユーザー一覧R U = $\emptyset$   
制限スキーマ一覧R S = $\emptyset$

【0071】例4に対し、ビュー名のみを指定し、ビュ

例6 複数ビューを退避、関連実表自動退避  
対象物のリスト=S 1. V 1, S 2. V 2  
対象範囲 =DOWN  
制限範囲指定 =なし  
ユーザー一覧U =鈴木、田中  
スキーマ一覧S =S 1, S 2  
ビュー一覧V =V 1, V 2  
実表一覧T =T 1, T 2  
権限一覧A =T 1 に関する田中の選択権  
プログラム一覧P = $\emptyset$   
制限ユーザー一覧R U = $\emptyset$   
制限スキーマ一覧R S = $\emptyset$

【0072】

例7 ビューを退避、参照している表を退避  
対象物のリスト=S 1. V 2  
対象範囲 =DOWN  
制限範囲指定 =なし  
ユーザー一覧U =鈴木、田中  
スキーマ一覧S =S 1, S 2  
ビュー一覧V =V 2  
実表一覧T =T 1, T 2  
権限一覧A =T 1 に関する田中の選択権  
プログラム一覧P = $\emptyset$   
制限ユーザー一覧R U = $\emptyset$   
制限スキーマ一覧R S = $\emptyset$

【0073】

例8 ビューを退避、関連表を全て自動退避  
対象物のリスト=S 1. V 2  
対象範囲 =ALL  
制限範囲指定 =なし  
ユーザー一覧U =鈴木、田中

スキーマ一覧 S	= S 1, S 2, S 3
ビュー一覧 V	= V 2, V 3
実表一覧 T	= T 1, T 2
権限一覧 A	= T 1 に関する田中の選択権
プログラム一覧 P	= $\emptyset$
制限ユーザー一覧 R U	= $\emptyset$
制限スキーマ一覧 R S	= $\emptyset$

#### 【0074】

##### 例9 関連の複雑なビューを退避

対象物のリスト	= V 3
対象範囲	= DOWN
制限範囲指定	= なし
ユーザー一覧 U	= 鈴木、田中
スキーマ一覧 S	= S 1, S 2, S 3
ビュー一覧 V	= V 2, V 3
実表一覧 T	= T 1, T 2
権限一覧 A	= T 1 に関する田中の選択権
プログラム一覧 P	= $\emptyset$
制限ユーザー一覧 R U	= $\emptyset$
制限スキーマ一覧 R S	= $\emptyset$

#### 【0075】

##### 例10 ユーザの所有物を全て退避

対象物のリスト	= 鈴木
対象範囲	= DOWN
制限範囲指定	= なし
ユーザー一覧 U	= 鈴木
スキーマ一覧 S	= S 1
ビュー一覧 V	= V 1
実表一覧 T	= T 1
権限一覧 A	= $\emptyset$
プログラム一覧 P	= $\emptyset$
制限ユーザー一覧 R U	= $\emptyset$
制限スキーマ一覧 R S	= $\emptyset$

#### 【0077】

##### 例12 スキーマ下全てと表を退避

対象物のリスト	= S 1, S 2, T 2
対象範囲	= DOWN
制限範囲指定	= なし
ユーザー一覧 U	= 鈴木、田中
スキーマ一覧 S	= S 1, S 2
ビュー一覧 V	= V 1
実表一覧 T	= T 1, T 2
権限一覧 A	= $\emptyset$
プログラム一覧 P	= $\emptyset$
制限ユーザー一覧 R U	= $\emptyset$
制限スキーマ一覧 R S	= $\emptyset$

#### 【0078】

#### 【0076】

##### 例11 あるスキーマの下を全て退避

対象物のリスト	= S 1
対象範囲	= DOWN
制限範囲指定	= なし
ユーザー一覧 U	= 鈴木
スキーマ一覧 S	= S 1
ビュー一覧 V	= V 1
実表一覧 T	= T 1
権限一覧 A	= $\emptyset$
プログラム一覧 P	= $\emptyset$
制限ユーザー一覧 R U	= $\emptyset$
制限スキーマ一覧 R S	= $\emptyset$

##### 例13 プログラムと関連物を退避 対象物のリスト = P R O G 2

対象範囲 = DOWN  
 制限範囲指定 = なし  
 ユーザー一覧 U = 鈴木、田中  
 スキーマ一覧 S = S1、S2、S3  
 ピュー一覧 V = V1、V2、V3  
 実表一覧 T = T1、T2  
 権限一覧 A = T1に関する田中の選択権  
 プログラム一覧 P = PROG2  
 制限ユーザー一覧 RU =  $\emptyset$   
 制限スキーマ一覧 RS =  $\emptyset$

#### 【0079】

例 14 プログラムと表を退避  
 対象物のリスト = PROG1、T1  
 対象範囲 = DOWN  
 制限範囲指定 = なし  
 ユーザー一覧 U = 鈴木、田中  
 スキーマ一覧 S = S1、S2  
 ピュー一覧 V = V1、V2  
 実表一覧 T = T1、T2  
 権限一覧 A = T1に関する田中の選択権  
 プログラム一覧 P = PROG1  
 制限ユーザー一覧 RU =  $\emptyset$   
 制限スキーマ一覧 RS =  $\emptyset$

#### 【0080】

例 15 スキーマ制限付でプログラムを退避  
 対象物のリスト = PROG2  
 対象範囲 = DOWN  
 制限範囲指定 = S1  
 ユーザー一覧 U = 鈴木  
 スキーマ一覧 S = S1  
 ピュー一覧 V = V1  
 実表一覧 T = T1  
 権限一覧 A =  $\emptyset$   
 プログラム一覧 P = PROG2  
 制限ユーザー一覧 RU =  $\emptyset$   
 制限スキーマ一覧 RS =  $\emptyset$

#### 【0081】

例 16 ユーザ制限付でビューを退避  
 対象物のリスト = V3  
 対象範囲 = DOWN  
 制限範囲指定 = 鈴木  
 ユーザー一覧 U = 鈴木  
 スキーマ一覧 S = S1  
 ピュー一覧 V = V1  
 実表一覧 T = T1  
 権限一覧 A =  $\emptyset$   
 プログラム一覧 P = PROG1  
 制限ユーザー一覧 RU =  $\emptyset$   
 制限スキーマ一覧 RS =  $\emptyset$

【0082】次に、図2に保存媒体の内容の例を示す。

図2において、50は置換情報、60は定義情報、80は依存情報、100はデータ本体、110はジャーナル、120はプログラム群である。

【0083】置換情報50にはユーザ名の置換情報が記録されている。この例では、置換情報各々は\$ではじまる仮の名前で表されている。この例においては、ユーザ1が鈴木であり、ユーザ2が田中であることを示しており、この置換情報を置換しなければ、ユーザ1は鈴木、ユーザ2は田中として復元、或いは移行できる。しかし、移行先のシステムに既にユーザ名鈴木、又はユーザ名田中が存在している場合にはこの置換情報を用いることにより、例えばユーザ1を鈴木ではなく山田というユーザ名に変更することが可能になる。すなわち、以後に出てくる\$U1という変数は全て山田に置き換えられてインストールされることになる。

【0084】次に、定義情報60は定義情報を生成するための情報がSQL文により、記述されている。61においてはユーザ1が生成され、62においてはユーザ2が生成されることを示している。63においては、次の64のスキーマ作成者をユーザ1に設定する。そして64においては、63において設定されたユーザ1の所有物としてスキーマS1を生成する。同様に、65から67においてはユーザ2に対してスキーマS2、S3が生成される。また68においては以下のSQL文の実行者をユーザ1に設定する。69においては実表T1が生成される。70においては実表T1を参照する権限をユーザ2に与えると共に、ユーザ2はその参照権限を他のユーザ

一に与えるオプションを持っていることを宣言している。71においては実表T1から列t11が100以上のものだけをセレクトしてビューV1を生成することを定義している。次に、72～75においてはユーザ2に対してテーブル2を生成し、実表T1と実表T2からビューV2を生成し、更に、ビューV2からビューV3を生成することを記述している。

【0085】次に、依存情報80は、退避したデータベースシステムの定義情報から抽出された依存関係を示しているものである。この例においては81、82においてユーザ1及びユーザ2は何にも依存しないことを示している。また83においてはスキーマS1はユーザ1の存在が必要であることを示している。84と85においてはスキーマS2、S3はユーザ2の存在が必要であることを示している。即ち、スキーマ1を生成する場合には、ユーザ1が先に存在していなければならぬ。同様に、スキーマS2、S3を生成する場合にはユーザ2が存在しなければ生成できないことを示している。次に、86においては実表T1はスキーマS1の存在が必要であることを示している。87においては実表T2はスキーマS2の存在が必要であることを示している。88においてはビュー1は実表T1とスキーマS1の存在が必要であることを示している。89においてはビューV2が実表T1、T2及び、スキーマS2の存在が必要であるとともに、ユーザ1からユーザ2に対する実表T1に関する選択権が必要であることを示している。90においてはビューV3はビューV2とスキーマS3の存在が必要であることを示している。次に、89で示したユーザ1からユーザ2への実表T1に関する選択権は、ユーザ1とユーザ2と実表T1が存在することが必要であることを示している。92においてはプログラムP1はビューV2の存在が必要であることを示している。また、93においてはプログラムP2は実表T2が存在が必要であることを示している。94においては、プログラムP3はビューV1とビューV3の存在が必要であることを示している。

【0086】これらの依存情報は、従来例で説明した定義情報の一覧情報から抽出することが可能である。例えば、81、82はユーザ一覧情報を参照することによって抽出できる。また83～85はスキーマ一覧情報を参照することにより抽出することが可能である。また86、87は表一覧情報を参照することにより抽出可能である。また88～89は表一覧情報と、ビュー定義一覧及びビュー対表の参照関係及び権限一覧を参照する事により、抽出可能である。また、91は権限一覧を参照する事により抽出可能である。また92～94は手順対表参照関係から抽出することが可能である。

【0087】データ本体100はデータベースの実際のデータをSQL文のインサートコマンドにより記述したものであり、101は実表T1のデータ挿入コマンドで

あり、102は実表T2のデータ挿入コマンドである。

【0088】次にジャーナル110にはジャーナルの通常のファイルの形式の履歴データが存在するが、ここでは前述した退避すべき対象物に関する履歴情報が、表名という識別子により抽出されて記述される。例えば、この例においては、実表T1とT2が退避されるため、ジャーナル110には実表T1とT2という識別子がついた履歴データのみが、ここに付加されることになる。

【0089】次にプログラム群120には通常のファイル形式のプログラムが付加されるが、例えばこのプログラムをプログラムA、B、C、Dの4つのロードモジュールであるとするならば、このプログラムはCOBOL等の一般的のプログラム処理言語で記載されており、そのプログラム言語の一部にSQLで記述された手順がソースコードの形式で存在すると共に、その実行形式の手順を呼ぶためのコール文が存在している。この例においては、手順としてP1、P2、P3が退避べき対象物として特定されるが、手順P1、P2、P3は定義情報内にある実行形式の手順が保存されるのではなくCOBOLプログラムのロードモジュールA、B、C、D内にある手順P1、P2、P3のソースコードがプログラム群の一部として保存される。定義情報内にある実行形式の手順P1、P2、P3の保存はされない。

【0090】このように、通常プログラムに対応するロードモジュール中にデータベース処理言語の原プログラムを保持し、移動先のデータベース処理システム上でこのロードモジュールを実行し、データベース処理手順の識別子を用いて、移動先のデータベース処理システム上のデータベース処理手順を登録するカタログ領域にあるデータベース処理手順を検索し、これが存在しないことにより、ロードモジュール中のデータベース処理言語のプログラムを動的に再コンパイルし、これを移動先のシステムのデータベース処理手順を登録するカタログ領域に移動先のロードモジュールのファイル識別子と、データベース処理手順相対番号により新たに登録し、再び該データベース処理部の実行を継続する。この際、データベース処理手順の識別子は、カウンタなどの値ではなく、当該アプリケーションの走行しているシステム上で実行時に採取したロードモジュールのファイル識別子を用いる。従って、このような移植動作によってもデータベース処理手順の一意性が保証されることになる。即ち、埋め込まれたデータベース処理言語により記述されたデータベース処理部には、データベース処理アプリケーション実行時に得た当該アプリケーションプログラムの通常プログラム部に対応するロードモジュールのオペレーティングシステム上でのファイルの識別子と、当該データベース処理部の当該アプリケーションプログラム中での相対番号の値の組からなる識別子を与え、実行時に通常プログラムがデータベース処理が必要と成了った場合には、実行時に得た自ロードモジュールのオペレーテ

イングシステムのファイル識別子と、必要とするデータベース処理部のアプリケーション内相対番号を組として、データベース処理手順を登録したカタログ領域を検索し、これにより通常プログラム部から登録された所要のデータベース処理部を呼出す。

【0091】3) 退避移行復元手順生成装置の動作  
まず、渡された対象物指定を以下のように解釈し、対象物の一覧を得る。尚、以下で(\*)の部分は、退避の場合にはデータベースのカタログ情報から識別が可能であり、また復元の場合には依存情報から識別が可能である。

【0092】(1) ユーザー一覧U = $\phi$  : 指定されたユーザ全体を記憶する内部変数  
スキーマ一覧S = $\phi$  : 指定されたスキーマ全体を記憶する内部変数

ビュー一覧V = $\phi$  : 指定されたビュー全体を記憶する内部変数  
実表一覧T = $\phi$  : 指定された実表全体を記憶する内部変数  
権限一覧A = $\phi$  : 指定された権限全体を記憶する内部変数  
プログラム一覧P = $\phi$  : 指定されたプログラム全体を記憶する内部変数

【0093】(2) 指定された各対象物に対して指定されているものが以下のような場合、それぞれの対象物を次のように決定する。

なお、以下に用いた記号 $\cup$ はその記号左側に記載した集合に、その記号右側に記載したものと合併する事を示す記号である。

a) ユーザ名が指定されている場合

ユーザー一覧U = ユーザー一覧U  $\cup$  このユーザ名。  
スキーマ一覧S = シーマ一覧S  $\cup$  当該ユーザが所有している全てのスキーマ

実表一覧T = 実表一覧T  $\cup$  これらスキーマ中の全ての実表

(\*)

ビュー一覧V = ビュー一覧V  $\cup$  これらスキーマ中の全てのビュー

(\*)

b) スキーマ名が指定されている場合

ユーザー一覧U = ユーザー一覧U  $\cup$  当該スキーマを所有するユーザ

(\*)

スキーマ一覧S = シーマ一覧S  $\cup$  当該スキーマ

実表一覧T = 実表一覧T  $\cup$  当該スキーマ中の全ての実表

(\*)

ビュー一覧V = ビュー一覧V  $\cup$  当該スキーマ中の全てのビュー

(\*)

c) スキーマ名、表名が指定されている場合

スキーマ一覧S = シーマ一覧S  $\cup$  この表が所属するスキーマ名

(\*)

とする。更にこれらをこの表が実表であれば

(\*)

実表一覧T = 実表一覧T  $\cup$  この表

(\*)

そうでなければ

(\*)

ビュー一覧V = ビュー一覧V  $\cup$  この表

d) 権限が指定されている場合

ユーザー一覧U = ユーザー一覧U  $\cup$  当該権限を付与しているユーザ  $\cup$  当該権限を受領しているユーザ

(\*)

当該権限の対象物に対し、上記c)を実行する。

権限一覧A = 権限一覧A  $\cup$  当該権限

e) プログラム名が指定されている場合

当該プログラムが参照している表に対し、上記c)を実行する。

プログラム一覧P = プログラム一覧P  $\cup$  当該プログラム

上記の(\*)について、カタログからこれを得る場合については後述のデータベース及びカタログアクセス装置の説明の際に述べる。また、依存情報からこれを得る場合には、以下のようになる。

例1 : あるスキーマに属する表一覧を得る。

= このスキーマに依存するビュー及び表を取り出す。

図2に示す例では、S1に属する表一覧は86と88からS1に依存するT1とV1とわかる。

## 例2：あるビューが参照している表一覧

=このビューが依存するビュー及び実表を取り出す。  
図2に示す例では、ビューV2が参照する表は89のV2の依存部を見て、T1、T2とわかる。

【0094】(3) 対象範囲指定がONLYとTOWNとALLの場合の対象物の特定は、以下のようにして行う。

a) ONLYの場合、指定された対象物に対して、以下の確認を行なう。

- ・実表一覧T、ビュー一覧Vにある実表、ビューが所属するスキーマがスキーマ一覧S中にあるか？

- ・スキーマ一覧S中のスキーマの所有者がユーザー一覧Uにあるか？

- ・ビュー一覧Vの参照しているビュー、実表がビュー一覧V、実表一覧Tにあるか？

- ・ビュー一覧Vの生成に必要な権限が権限一覧Aにあるか？

以上の条件を満たさない場合は、エラーとする。

b) DOWNの場合は、以下のようにして対象物の特定を行う。

これは、以下の退避復元移行手順生成装置のうきと同じであるが、説明のため、ここに記述する。

- ・現在のビュー一覧Vの各ビューVについて、以下の特定を行う。

ビューVが参照している各表tについて、

表tがビューの場合：ビュー一覧V=ビュー一覧V U 表t

表tをビューVとしてこれを再帰的に繰り返す。

表tが実表の場合：実表一覧T=実表一覧T U 表t

- ・ビュー一覧V、実表一覧Tが所属するスキーマでまだスキーマ一覧Sにないものをスキーマ一覧Sに加える。

- ・スキーマ一覧Sにあるスキーマでまだその所有者がユーザー一覧Uにないものをユーザー一覧Uに加える。

これにより、全ての対象物を指定せずとも、a) ONLYの場合の条件が自動的に満たされ、対象物の内容は閉じたものとなる。

例1：ユーザー一覧U=田中、スキーマ一覧S=S2、ビュー一覧V=V2、実表一覧T=の場合、

V2が参照している実表T1、T2を実表一覧Tに加える。スキーマ一覧SにS1を、ユーザー一覧Uに鈴木を加える。

例2：ユーザー一覧U=田中、スキーマ一覧S=S3、ビュー一覧V=V3、実表一覧T=の場合、

V3が参照しているビューV2をビュー一覧Vに加える。スキーマ一覧SにS2を加える。

更にV2が参照している実表T1、T2を実表一覧Tに加える。スキーマ一覧SにS1を、ユーザー一覧Uに鈴木を加える。

c) ALLの場合は、以下のようにして対象物の特定を行なう。

- ・DOWNの場合の手順を実行する。

- ・B=実表一覧T

- ・D=

- ・L：現在のBの各表tについて

- a) もし、全てのtについてこれを参照しているビューがなければ終わり

- b) そうでなければtを参照している各ビューVについて

- ・権限一覧A=権限一覧A U tに関してvの所有者に出されている権限

- ・ビュー一覧V=ビュー一覧V U v

- ・vが参照している表が全てB中にある場合：

B=B U v

D=D U v

- ・B=D、D=

- ・Lに戻る。

これらDOWNとALLの場合を図示すると図3のようになる。

【0095】(4) 制限範囲指定で指定されたユーザ名群を制限ユーザー一覧RUに入れる（なければ制限ユーザー一覧RUは）制限範囲指定で指定されたスキーマ名群を制限スキーマ一覧RSに入れる（なければ制限スキーマ一覧RSは）。

【0096】(5) ユーザー一覧U=ユーザー一覧U U 制限ユーザー一覧RU（ユーザー一覧Uと制限ユーザー一覧RUの交わり）

スキーマ一覧S=スキーマ一覧S U 制限スキーマ一覧RS（スキーマ一覧Sと制限スキーマ一覧RSの交わり）

【0097】(6) 以上で定まった実表一覧T、ビュー一覧Vからスキーマ一覧Sにあるスキーマ中でない実表、ビュー表を除く。

【0098】(7) 復元、移行の場合には、これらユーザー一覧U、スキーマ一覧S、ビュー一覧V、実表一覧T、権限一覧A、制限ユーザー一覧RU、制限スキーマ一覧RSを、指定されたパラメータとともに制御部に渡す。

【0099】(8) 退避の場合には以下の手順を実行する。

- 1) ユーザー一覧Uで指定されたユーザを生成するCREATE USER文を生成し、定義情報とする。この際、ユーザ名は個々に異なる置き換え可能な変数とし、この変数名とユーザ名自身を組として置換情報として順に記憶する。

- 2) 指定されたユーザー一覧U及び指定されたスキーマ一覧Sを基に、これらスキーマを生成するCREATE SCHEMA文を生成し、定義情報として記憶する。

- 3) これらユーザ、スキーマに関連する表、ビュー、アクセス権を生成、定義するCREATE TABLE、CREATE VIEW、GRANT文を生成し、定義情報として記憶する（詳細は後述する）。

4) 上記において判明している依存関係を依存情報として記憶する。依存関係は以下のようにして識別される。

- ・ユーザー一覧U中のユーザ名は何にも依存しない。
- ・スキーマ一覧S中のスキーマ名はその所有者のユーザに依存するとして依存情報を生成する。
- ・実表一覧T中の実表はそれが存在するスキーマに依存するとして依存情報を生成する。
- ・ビュー一覧V中のビューは、ビュー一覧Vが参照している実表またはビューとそれに関する権限、及びビュー一覧Vが存在するスキーマに依存するとして依存情報を生成する。
- ・権限は、権限付与者、権限受領者、権限対象の表に依存するとして依存情報を生成する。
- ・プログラムはプログラムが参照している表に依存するとして依存情報を生成する。

【0100】次に回避復元手順生成装置が行う表、ビュー、権限の保存の手順に付いて説明する。表、ビューは相互に関連があり、またビューを生成する際には権限が必要となる。従って、ビュー生成の文を実行する前にはビューが参照している表生成のCREATE TABLE文とビューが仮定している権限を定義するGRANT文が実行されてなければならない。従って、表、ビュー、権限の保存の大まかな手順は以下のようになる。

(1) まず、該当する表をすべて生成する。この際に各表を生成する度に表に関して付与されている権限を定義する。

(2) 次に、これら表のみを参照しているビューを生成する。この際に各ビューを生成する度にビューに関して付与されている権限を定義する。

(3) 次に、これら表、及びビューのみを参照しているビューを生成する。この際に各ビューを生成する度にビューに関して付与されている権限を定義する。

(4) 現在までに生成された表及びビューを参照しているビューがなくなると停止する。

【0101】以上の過程において、権限に対するGRANT生成の際、複数のGRANT文が生成される場合がある。例えば、表T1を伏見が作成し、この権限をWITHEGRANT OPTION付きで安藤に渡す。安藤は岩崎に更に渡す。この場合、伏見→安藤、安藤→岩崎の順にGRANT文を生成しないと正しく実行できない。この順序を正しく生成する手順が必要となる。以下に、もっとも一般的な形として上記手順により実表一覧が得られ、かつ対象範囲指定でALLが選択された場合に、CREATE TABLE文、CREATE VIEW文、GRANT文が正しい順で生成される手順の一例を示す。尚、CREATE USER文、CREATE SCHEMA文は各々ユーザー一覧で、スキーマ一覧Sからそのまま生成できる為、その説明は省略する。

【入力】

実表一覧T

: 対象となる表一覧

【出力】

CREATE TABLE, CREATE VIEW, GRANT文の列、

【手順:】

(1) ハッシュ表Bを空にする。Bは現在までにCREATEを生成したすべての表、ビューを記憶する。

(2) ハッシュ表Dを空にする。Dは、現在までにCREATEを生成したビュー、表の内、まだ上位のビューで参照されている可能性のあるもの全体を示す。

(3) 実表一覧T中の各表tに対し、tがスキーマ一覧Sのいづれの要素のスキーマにもなければ何もしない。そうでなければ表tに対するCREATE TABLE文を生成する。表tをスキーマ名とともにハッシュ表Bに登録する。表tをスキーマ名とともにハッシュ表Dに登録する。表tをオブジェクトとするGRANT文を下記のGRANT文生成手順で生成する。

(4) ここをLOOPラベルとする。ハッシュ表Cを空にする。Cは、Dの要素であるビュー、表を参照しているビューの全体を示す。ハッシュ表Dの各表dに対し、表dを参照している各ビューvに対し、以下a)を繰り返す。

a) もし、ビューvがスキーマ一覧Sの要素のいづれかのスキーマにあればビューvをハッシュ表Cに登録する。ハッシュ表Cが空であればDを空にして終り。そうでなければ、ハッシュ表Dを空にする。ハッシュ表Cの各表cに対して、以下a)、b)を繰り返す。ビューvが参照している各ビューまたは表tvに対し、

a) もしtvがハッシュ表Bにあればcontinue

b) もしtvがハッシュ表Bにあればbreak

もしvが参照しているすべてのビューまたは表がハッシュ表Bに存在すれば(つまりcontinueで上を終了すれば)、

a) vに関するCREATE VIEW文を生成する。

b) vをハッシュ表Bに登録する。

c) vをハッシュ表Dに登録する。

LOOPラベルに戻る。

【0102】以上の手順を図2に示す具体例を用いて説明する。先ず図4に示されるようにハッシュ表B、C、Dが用意されているものとする。まず、スキーマS1から表T1が選択され、実表T1が生成され、実表T1がスキーマ名と共にハッシュ表Bとハッシュ表Dに登録される。そして、この実表T1をオブジェクトとするGRANTが生成される。ここまで時点で図2に示す68、69、70までの定義情報が生成される。次に、この実表T1を参照しているビューV1が存在しているので、これをハッシュ表Cに登録する。この時点ではハッシュ表B、C、Dは図5に示すような状態になる。

【0103】次に、ハッシュ表Dが空にされ、ハッシュ表Cに登録されているビューV1が参照しているビューまたは表(この場合には表T1)が既にハッシュ表Bに

登録されているため、ビューV1に関するCREATE文を生成する。この時点で図2の71が生成される。そして、ビューV1をハッシュ表B及びハッシュ表Dに登録する。この時点で各表は図6に示されるような状態になる。尚、図2の70については後述する。

【0104】次に、前述した手順のループラベルの部分に戻ることになる。ハッシュ表Cを空にし、ハッシュ表Dにある表V1に対して表V1を参照しているビューがあるかどうかをチェックする。この時点ではビューV1を参照しているビューは存在していないため、ハッシュ表Cに登録するものは何もなく、ハッシュ表が空であるためこのループを終了する。

【0105】次にスキーマS2が持つ実表T2に対して同様の処理が行われる。図7に示すように、実表T2がスキーマ名と共にハッシュ表B及びハッシュ表Dに登録される。この時点で、各表は図7に示すような状態になる。次に、この実表T2に対するGRANT文を生成することになるが、この実表T2に対しては権限の付与がないため、何も生成されない。以上の時点までで、図2に示す72と73が生成されたことになる。

【0106】次にハッシュ表Cを空にし、ハッシュ表Dに登録された実表T2を参照しているビューがあるかどうかをチェックする。この場合にはビューV2が存在しているため、ビューV2をハッシュ表C2に登録する。この時点で各表は図8に示すような状態になる。

【0107】次にハッシュ表Dを空にし、ハッシュ表Cに登録されたビューV2が参照している各ビューまたは表が、ハッシュ表Bに登録されているかどうかをチェックする。ビューV2は実表T1とT2を参照しているが、このT1とT2は共に既にハッシュ表Bに登録されている。従ってビューV2に関するCREATE文を作成する。この時点で図2に示した74が生成される。そして、ビューV2をハッシュ表Bとハッシュ表Dに登録する。この時点で各表は図9に示したような状態になる。そしてループラベルに戻る。ループラベルに戻るとハッシュ表Cを空にし、ハッシュ表Dに登録されている表に対してその表を参照しているビューが有るかどうかをチェックする。ビューV2に対しては、ビューV3が参照しているためビューV3をハッシュ表Cに登録する。この時点で各表は図10に示すような状態になる。ビューV3は、ビューV2のみを参照しており、このビューV2はすでにハッシュ表Bに登録されているため、このビューV3に対するCREATE文を生成する。この時点で図2の75が生成される。そしてビューV3をハッシュ表Bとハッシュ表Dに登録する。この時点で各表は図11に示すような状態になる。ここで再びループラベルに戻るが、次にはこのビューV3を参照するビューが存在しない為、ハッシュ表Cに新たに登録するビューは存在せず、ハッシュ表が空になるためこの処理を終了する。このようにして前述した手順を用いることに

より、表及びビューを矛盾なく生成することが可能になる。

【0108】次に、GRANT文生成の手順について説明する。

GRANT文生成の手順（入力：対象物名、所有者名）

- (1) ハッシュ表Gを空にする。
- (2) Gに"所有者名"を登録する。
- (3) ここをLOOPラベルとする。
- (4) Gが空ならばここでおわり。

そうでなければHを空にする。

G内の各人間gに対し、付与者がg、対象物が"対象物名"である権限すべてに関して、GRANT文を生成する。

権限がWITH GRANT OPTION付きであり、かつ権限受領者がユーザー範囲Uの要素であれば、Hにその権限の権限受領者を登録する。

G = Hとする。

LOOPラベルに戻る。

【0109】次に、上記GRANT文生成手順の具体例について説明する。このGRANT文生成手順は前述した表、ビューの保存手順を生成する過程に呼ばれるものであり、ここでは図2に示した70のGRANT文が生成される場合について説明する。図2の70は\$U1が\$U2に対して選択権を与えていたものであるが、図12に示すように、ハッシュ表Gに先ず\$U1として所有者名鈴木が登録される。次にこの鈴木が持っている権限全てに関してGRANT文を生成する。このGRANT文の生成は、図24に示した権限一覧を参照することにより行われる。権限一覧にはシステムが権限を付与した場合と、特にユーザが他のユーザに権限を付与した場合が存在するが、ここでシステムが付与した権限は無視され、あるユーザが他のユーザに付与した権限のみが選択される。この例においてはユーザ鈴木がユーザ田中に対して選択権を与えていたため、ユーザ鈴木からユーザ田中に対して選択権を与えるGRANT文が生成される。また、この場合は、権限を再付与することができるため、表Hにその権限を再付与することができるユーザ名田中を登録する。この時点で図13のような状態になる。次に、この表Hをハッシュ表Gに置き換える。即ち、図14のようにハッシュ表Gには田中が登録される。この田中が持つ権限を前述したと同様に権限一覧の中から検索し、もし田中が他のユーザに対して権限を与えていた場合には、その権限全てに対して再びGRANT文が生成されることになるが、この例では田中が他のユーザに権限を与えていたことは、権限一覧には登録されていないため、GRANT文は生成されない。このように、ユーザ1が鈴木である場合にはGRANT文が1つ生成される。

【0110】4) データベース及びカタログアクセス装置の動作

データベースの内容自身、及びカタログの内容をアクセスする装置である。この部分は、データベース中核に特別なインターフェースを設けて実現することができるが、SQLなどでは、この部分自身もSQLで記述できる。まずデータベースのアクセスについて説明する。例えば、指定した実表Tの内容を取り出すには、

```
SELECT * FROM T
なるSQL文を実行すればよい。また、指定した内容を
```

```
INSERT INTO T1
```

次にカタログのアクセスについて説明する。例えば、退避移行復元手順生成装置で必要とされるカタログ情報は以下のようにしてSQLによりカタログから得られる。

当該ユーザuが所有している全てのスキーマ  
SELECT スキーマ名 FROM スキーマ情報一覧

WHERE 所有者=u

スキーマs中の表tを参照しているビューの一覧

SELECT ビュー名 FROM ビュー対表の参照関係

WHERE 参照表名=t AND 参照表スキーマ名=s

以上のようにすることにより通常のSQLプログラムにより通常のSQLのインターフェースで必要な情報が採取

```
CREATE VIEW V1 (v11, v12)
AS SELECT * FROM T1 WHERE t11 > 100
```

を実行することにより、ビュー表参照関係の情報や、スキーマ内に表が増加したことによるスキーマ内表数の列なども自動的に更新されることになる。即ち、これら情報は、退避の際に保存する必要がないことがわかる。

【0112】以上のように、この実施例は、データベースシステム中で開発された複数の業務から所要の業務に関連する定義情報を取り出し、これに加えて、業務に必要とされるデータ、業務プログラムのみをとりだし、業務パッケージとして作成し、これを別のマシン、システムに移行可能とすることを特徴とする。

【0113】また、定義情報の取りだしの際に、カタログ情報自身がSQLでアクセス可能な場合、カタログ情報の読みだしをSQLにより実行することにより特別なカタログ情報アクセス手段を不要とすることを特徴とする。

【0114】また、移行内容の内の定義情報の部分を、通常のSQLの定義情報定義のコマンド(CREATE TABLEなど)の履歴的に矛盾のない列に変換し、復元時には特殊なシステムインターフェースを不要とし、通常のSQLの定義コマンド群を指定された順に実行することにより復元が可能とする。また、データ自身もSQLのデータ投入コマンドにより変換し、移行時これを実行することにより復元が可能になる。更に、これがSQLという国際標準言語を用いるため、ベンダを超えてパッケージの移行が可能となる。例えば、上記のスキ

実表Tの内容にするには、

```
INSERT INTO T1
```

((1, AAA, 2), (2, bbb, 4), ...)

なるSQL文を実行すればよい。多くのSQLシステムでは更に高速の表の内容挿入インターフェースを持っているのでそれを用いてもよい。例えば、論理名'DEV/MT00'なる磁気テープの内容を実表Tの内容とするには、

```
FROM /DEV/MT00
```

できる。

【0111】また、復元時及び移行時には、対象となるシステムにおけるカタログ内容を復元するには、上記のようにSQLの定義系コマンドを実行することにより自

動的に復元できる。例えば移行先で、

```
CREATE TABLE T1 (
  t11 NUMERIC [4]
  t12 CHARACTER [8]
```

を実行することにより、表一覧、列一覧情報中に所要の情報が設定される。また、表一覧情報中の表毎の列の個数なども保存内容になくともこれらコマンドの実行により自動的に復元される。あるいは、

動的に復元できる。例えば移行先で、

```
CREATE TABLE T1 (
  t11 NUMERIC [4]
  t12 CHARACTER [8]
```

を実行することにより、表一覧、列一覧情報中に所要の情報が設定される。また、表一覧情報中の表毎の列の個数なども保存内容になくともこれらコマンドの実行により自動的に復元される。あるいは、

```
CREATE VIEW V1 (v11, v12)
AS SELECT * FROM T1 WHERE t11 > 100
```

マ中の表の数などは、移行先で表定義のSQL文を順次実行することによりシステムにより通常の方法で自動的にこの部分を変更するため、関連する情報の移行が不要となる。また、特殊なインターフェースが不要となる。

【0115】また、ユーザ名、データが格納されるシステム上の位置など、移行先のシステムで衝突の可能性のあるものを置き換えることとし、移行時にはこれらを適宜置き換えるだけでパッケージのインストールが可能なことを特徴とする。

【0116】また、SQLの手続き部分は保管せず、移行先で実行を行なった際にカタログ内にこの手続きがないことによりその場で再コンパイルを行ない、その結果を登録することにより、SQLの手続き部を保管するための記憶を節約し、更に移行先での最適化パラメタなどに応じた業務プログラム移行を可能とすることを特徴とする。

【0117】また、データベース本体を退避/復元せず、復元時にデータベース定義及びその記憶領域のみを確保し、退避のための記憶領域を節約する。更に、これにより業務動作に必要な枠組みをパッケージとしてインストールし、データベース内容は移行先で業務の運用が進むに連れて蓄積されることを可能とする。

【0118】また、この手順を従来のデータベースの単純な保管/復元手順にも応用し、パッケージ生成とデータベースの保管/復元を一つの手順で実行することを特

徵とする。

【0119】また、ジャーナルを保管する際には、ジャーナルの履歴レコードに表の識別子を付与しておくことによって、表単位の保管／復元を可能とすることを特徴とする。

【0120】実施例2. 次に、移行または復元の際には、退避されたデータをそのまますべて用いて、移行、または復元する場合に限らず、移行または復元の際に、移行／復元手順生成装置3により移行先のシステム環境にあった移行仕様や、復元すべき対象物の限定等の復元仕様を与えることができる。例えば、復元する実表とそれを参照しているビューを復元するのではなく、復元時に実表に対してビューを評価実行し、その結果を実表として復元することで、以降先でのディスク要領を減らし、また用途に応じたデータのみの復元、移行を可能とする。

【0121】また、復元しようとする実表に対し、SQL文による選択、除外を実行し、移行先または復元先で必要なデータのみを復元することを可能とする。また、データベース本体を退避している場合でも、移行仕様を与えることにより、データベース定義及びその記憶領域のみを確保することも可能である。これにより業務動作に必要な枠組みをパッケージとしてインストールし、データベース内容は移行先で業務の運用が進むに連れて蓄積されることを可能とする。また、復元する場合でも、復元仕様を与えることにより一部の実表のみを復元することが可能になる。また、ジャーナルに対しても、保存されたジャーナルから復元する実表に対するジャーナルのみを選択することが可能になる。このように、退避されたデータの一部を用いて移行または復元する場合は、退避されたデータの中に依存情報が存在しているからであり、この依存情報も参照することにより、退避されたデータの中から更に、一部のデータを矛盾無く抽出することが可能になる。

【0122】実施例3. 上記実施例においてはSQLを用いて説明したが、SQLを用いず他のデータベース処理言語を用いる場合においても、上記実施例と同様の効果を奏する。

【0123】実施例4. また、上記実施例1においてはデータベースシステム退避／移行／復元装置として、データベース及びプログラムの退避と移行と復元が1つの装置において出来る場合を説明したが、退避のみを行う装置、移行のみを行う装置、或いは復元のみを行う装置が別々に存在していても構わない。

【0124】

【発明の効果】以上のようにこの発明によれば、データベースシステムから指定した業務に必要な定義情報及びデータ及びプログラムを時間軸或いは相互関係に矛盾無く抽出することができ、これを用いてデータベースシステムを復元或いは移行する事が容易に出来るという効果

がある。

【0125】また、データベースの復元とデータベースの移行を同一の仕様により行うことができるため、データベースの復元と移行のための特別な処理手段を別々に必要としないという効果がある。

【0126】またデータベースシステムの退避をデータベース処理言語のコマンドの形に変換して退避するため、この退避したデータを復元する場合、或いは他のシステムに移行する場合には、データベース処理言語を実行できるシステムが備わっていればよく、特別な復元手段或いは、移行手段を必要としないという効果がある。

#### 【図面の簡単な説明】

【図1】この発明に係るデータベース及びプログラム退避／移行／復元装置のブロック図を示す図である。

【図2】この発明に係るデータベースシステム退避装置により退避される保存情報の一例を示す図である。

【図3】この発明に係るデータベースシステムの表とビューを退避する場合の退避すべき表とビューの特定方法を示す図である。

【図4】この発明に係る退避手順生成装置の表とビューの退避手順を説明する図である。

【図5】この発明に係る退避手順生成装置の表とビューの退避手順を説明する図である。

【図6】この発明に係る退避手順生成装置の表とビューの退避手順を説明する図である。

【図7】この発明に係る退避手順生成装置の表とビューの退避手順を説明する図である。

【図8】この発明に係る退避手順生成装置の表とビューの退避手順を説明する図である。

【図9】この発明に係る退避手順生成装置の表とビューの退避手順を説明する図である。

【図10】この発明に係る退避手順生成装置の表とビューの退避手順を説明する図である。

【図11】この発明に係る退避手順生成装置の表とビューの退避手順を説明する図である。

【図12】この発明に係る退避手順生成装置がGRANT文を生成する手順を説明するための図である。

【図13】この発明に係る退避手順生成装置がGRANT文を生成する手順を説明するための図である。

【図14】この発明に係る退避手順生成装置がGRANT文を生成する手順を説明するための図である。

【図15】従来のデータベースシステムのブロック図である。

【図16】従来のデータベースシステムのSQL手順の実行方式を説明するための図である。

【図17】データベースシステムのブロック図を示す図である。

【図18】ユーザー一覧情報の具体例を示す図である。

【図19】スキーマ一覧情報の具体例を示す図である。

【図20】表一覧情報の具体例を示す図である。

【図21】データベースシステムの列一覧情報を示す図である。

【図22】データベースシステムのビュー定義一覧を示す図である。

【図23】データベースシステムのビュー対表の参照関係を示す図である。

【図24】データベースシステムの権限一覧を示す図である。

【図25】データベースシステムのプログラム一覧を示す図である。

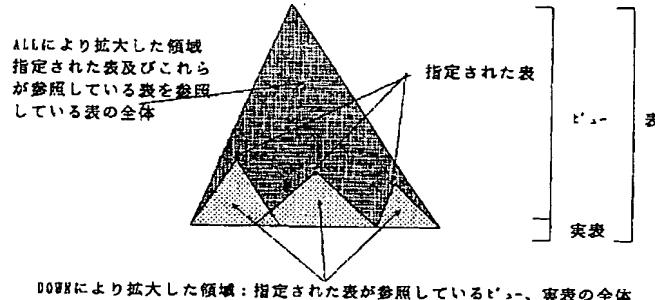
【図26】データベースシステムの手順一覧情報を示す図である。

【図27】データベースシステムの手順対表参照関係を示す図である。

【符号の説明】

- 1 コマンド解釈装置
- 2 制御部
- 3 退避移行復元手順生成装置
- 4 ビューSQL実行装置
- 5 保存媒体アクセス装置
- 6 データベース及びカタログアクセス装置
- 7 プログラムアクセス装置
- 8 保存媒体
- 9 データベースシステム
- 10 データベースシステム中核部
- 11 カタログ(定義情報)
- 12 データベース
- 13 ジャーナル
- 14 実行形式

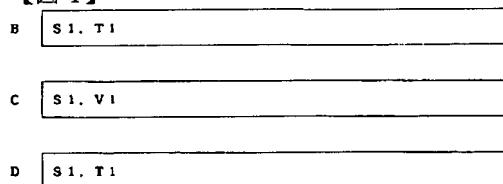
【図3】



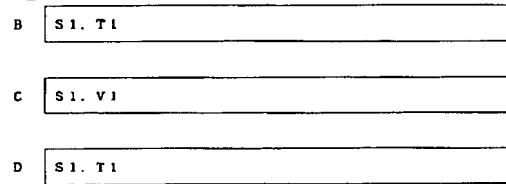
【図18】

ユーザ一覧情報	
ユーザ名	保有スキーマ数
鈴木	1
田中	2

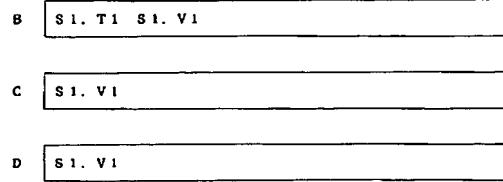
【図4】



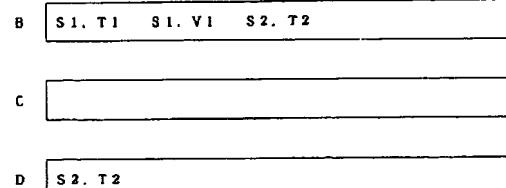
【図5】



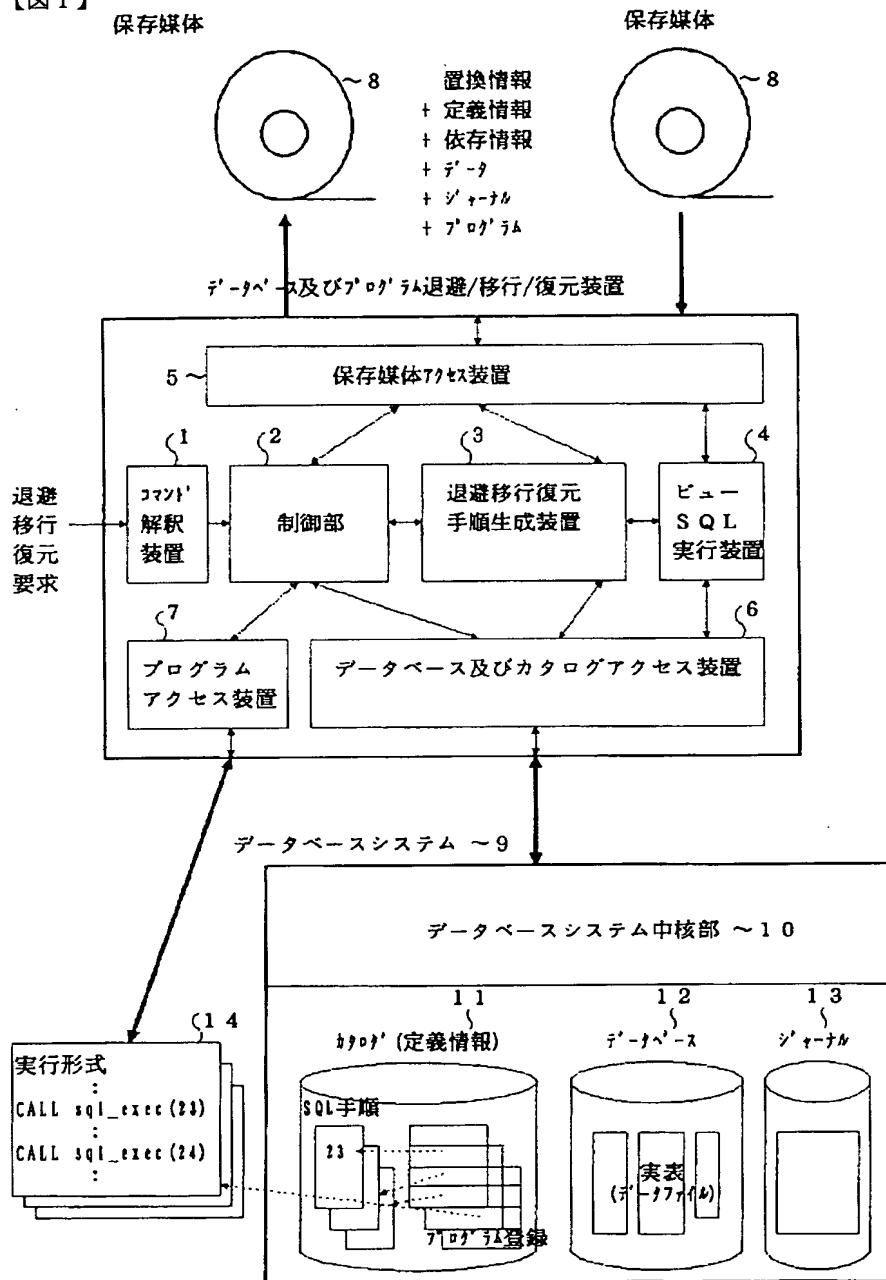
【図6】



【図7】



【图1】



【図2】

置換情報: ~50  
 \$U1: 鈴木 ~51  
 \$U2: 田中 ~52

定義情報: ~60

CREATE USER \$U1 ~61  
 CREATE USER \$U2 ~62

SET USER TO \$U1 ~63  
 CREATE SCHEMA S1 ~64

SET USER TO \$U2 ~65  
 CREATE SCHEMA S2 ~66  
 CREATE SCHEMA S3 ~67

SET USER TO \$U1 ~68

CREATE TABLE S1.T1 (I11 NUMERIC[4], I12 ....) ~69  
 GRANT SELECT ON S1.T1 TO \$U2 WITH GRANT OPTION ~70  
 CREATE VIEW S1.V1 (V11, V12) AS SELECT \* FROM S1.T1 WHERE I11 > 100 ~71

SET USER TO \$U2 ~72

CREATE TABLE S2.T2 (I21 NUMERIC[2], I22 ....) ~73

CREATE VIEW S2.V2 (V21, V22, V23) AS SELECT \*  
 FROM S1.T1, S2.T2 WHERE I11 = I23 ~74

CREATE VIEW S3.V3 (V31, V32, V33) AS SELECT \* FROM S2.V2 ~75

依存情報 ~80

\$U1: ~81  
 \$U2: ~82

S1: \$U1 ~83

S2: \$U2 ~84

S3: \$U2 ~85

T1: S1 ~86

T2: S2 ~87

V1: T1, S1 ~88

V2: T1, T2, S2, \$U1 → \$U2 の T1 に関する選択権 ~89

V3: V2, S3 ~90

\$U1 → \$U2 の T1 に関する選択権: \$U1, \$U2, T1 ~91

P1: V2 ~92

P2: T2 ~93

P3: V1, V3 ~94

データ本体 ~100

INSERT INTO T1 ((1, abcdef, 2), (2, dddddd, 3), (3, xxxxxx, 4), ....) ~101  
 INSERT INTO T2 ((1, abcdef, 2), (2, dddddd, 3), (3, xxxxxx, 4), ....) ~102

データーナル ~110

通常のファイルの形式 ~111

ログ群 ~120

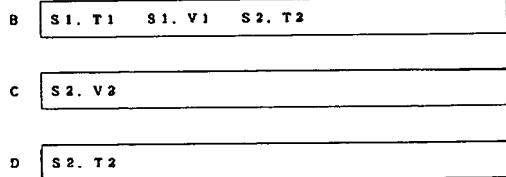
通常のファイルの形式 ~121

【図22】

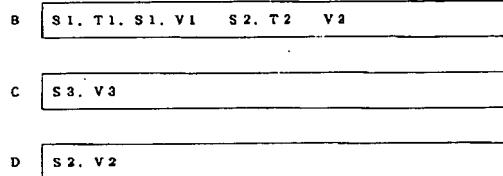
定義一覧

名	名	定義本体
V1	S1	'SELECT * FROM S1.T1 WHERE I11 > 100'
V2	S2	'SELECT I11, I12, I21 FROM S1.T1, S2.T2 WHERE I11 = I23'
V3	S3	'SELECT V21 FROM S2.V2'

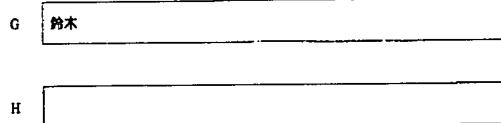
【図8】



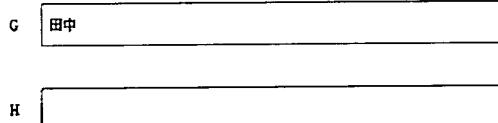
【図10】



【図12】



【図14】



【図19】

スキーマ一覧情報

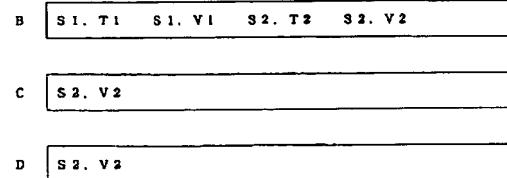
スキーマ名	所有者	格納表数
S 1	鈴木	1
S 2	田中	2
S 3	田中	1

【図25】

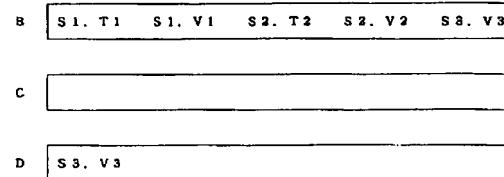
スキーマ一覧

スキーマ名	OS内スキーマ所在
PROG 1	/user/bin/prog1
PROG 2	/user/bin/prog2

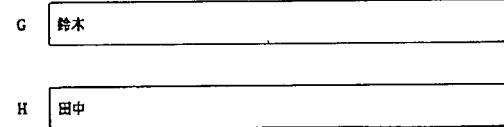
【図9】



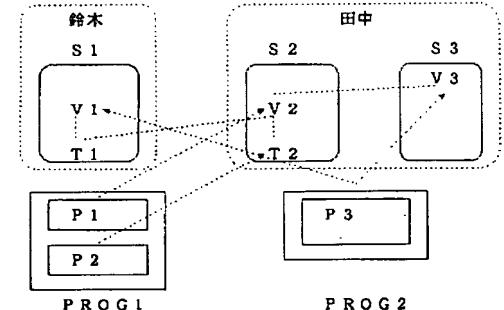
【図11】



【図13】

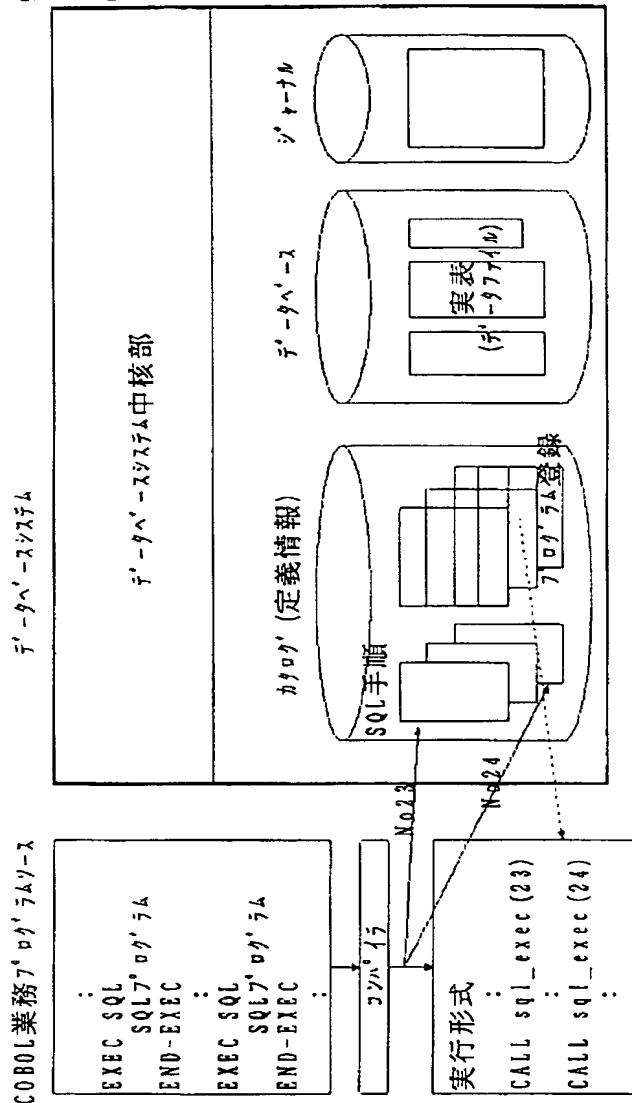


【図17】

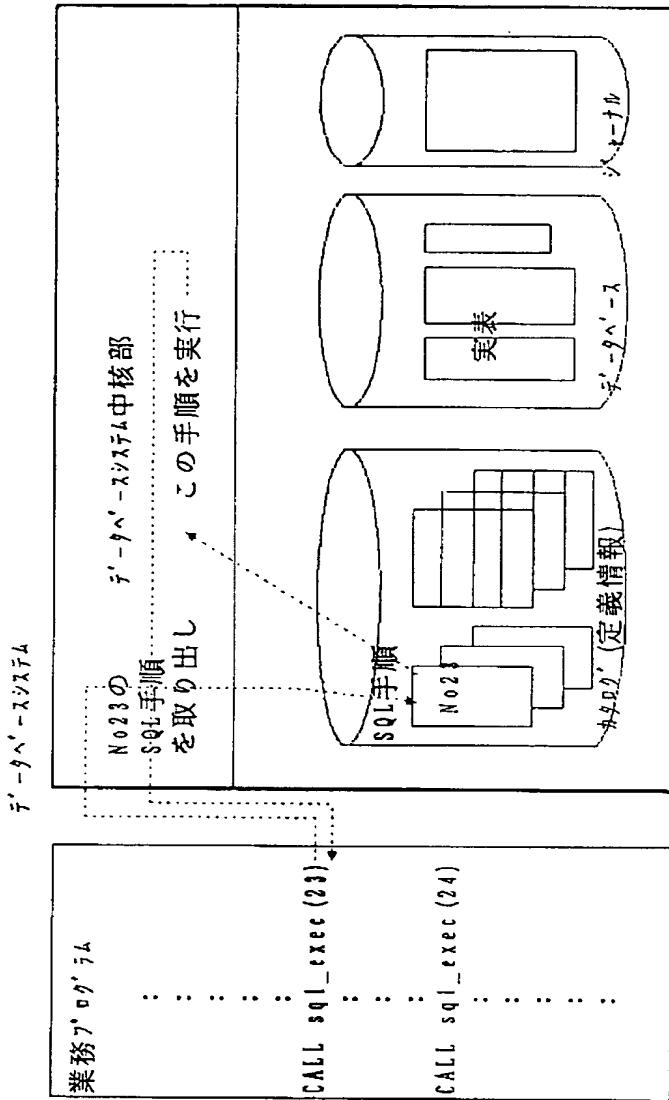
【図20】  
表一覧情報

表名	列数	スキーマ名	ヒュ-実表
T 1	2	S 1	実表
T 2	3	S 2	実表
V 1	2	S 1	ヒュ-実表
V 2	3	S 2	ヒュ-実表
V 3	3	S 3	ヒュ-実表

### 【图15】



【図16】



【図21】  
列一覧情報

列名	表名	スキーマ名	データ型	長さ
t11	T 1	S 1	数字	4
t12	T 1	S 1	文字	8
t21	T 2	S 2	数字	2
t22	T 2	S 2	文字	1 0
t23	T 2	S 2	数字	2
v11	V 1	S 1	数字	1 0
v12	V 1	S 1	文字	1 6
v21	V 2	S 2	数字	2
v22	V 2	S 2	文字	1 0
v23	V 2	S 2	数字	2
v31	V 3	S 3	数字	2
v32	V 3	S 3	文字	1 0
v33	V 3	S 3	数字	2

【図26】  
手順一覧情報

手順識別子	プロセス識別子	手順本体所在
P 1	PROG 1	/sql/plans/p1
P 2	PROG 1	/sql/plans/p2
P 3	PROG 2	/sql/plans/p3

【図27】  
手順対表参照関係

手順識別子	表名	対応名
P 1	V 2	S 2
P 2	T 2	S 2
P 3	V 1	S 1
P 3	V 3	S 3

【図23】  
ヒューリック表の参照関係

ヒューリック名	ヒューリックスキーマ名	参照表名	参照表スキーマ名
V 1	S 1	T 1	S 1
V 2	S 2	T 1	S 1
V 2	S 2	T 2	S 2
V 3	S 2	V 2	S 2

【図24】  
権限一覧

付与者	受領者	権限種類	対象	スキーマ名	再付与
SYSTEM	鈴木	選択権	T 1	S 1	YES
SYSTEM	鈴木	更新権	T 1	S 1	YES
SYSTEM	鈴木	削除権	T 1	S 1	YES
SYSTEM	鈴木	挿入権	T 1	S 1	YES
SYSTEM	鈴木	選択権	V 1	S 1	YES
SYSTEM	鈴木	更新権	V 1	S 1	YES
SYSTEM	鈴木	削除権	V 1	S 1	YES
SYSTEM	鈴木	挿入権	V 1	S 1	YES
SYSTEM	田中	選択権	T 2	S 2	YES
SYSTEM	田中	更新権	T 2	S 2	YES
SYSTEM	田中	削除権	T 2	S 2	YES
SYSTEM	田中	挿入権	T 2	S 2	YES
鈴木	田中	選択権	T 1	S 1	YES
SYSTEM	田中	選択権	V 2	S 2	YES
SYSTEM	田中	選択権	V 3	S 3	YES

【手続補正書】

【提出日】平成6年4月12日

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】全文

【補正方法】変更

【補正内容】

【書類名】 明細書

【発明の名称】 データベースシステム退避装置及びデータベースシステム復元装置及びデータベースシステム移行装置

【特許請求の範囲】

【請求項1】 以下の要素を有し、定義情報とこの定義情報により扱われるデータを記憶するデータベースシステムを退避するデータベースシステム退避装置

(a) 上記データベースシステムをアクセスするデータベースアクセス手段、(b) 退避すべきデータを保存データとして保存媒体に保存する保存媒体アクセス手段、

(c) 上記データベースシステムの退避命令を入力して解釈するコマンド解釈手段、(d) 上記データベースアクセス手段によりデータベースシステムの定義情報をアクセスすることにより、退避すべき対象物の退避すべき手順を生成する退避手順生成手段、(e) 上記コマンド解釈手段により解釈された退避命令に基づき、上記データベースシステムの中から退避すべき対象物を特定するとともに、上記退避手順生成手段を動作させ、退避手順生成手段により生成された退避すべき手順に従って、退避すべき対象物を上記データベースアクセス手段によりデータベースシステムから取り出し、上記保存媒体アクセス手段により保存媒体に保存する制御手段。

【請求項2】 上記コマンド解釈手段は、退避する対象物として、少なくともユーザ名、SQLのスキーマなどのデータの分類記述単位名、SQLの表やファイルなどのデータ名、SQLのビューなどの論理的データ名、プログラム名、権限名のいずれかひとつの対象物の指定と、この対象物に対する種類と範囲の指定がなされた退避命令を入力するとともに、上記制御手段は、退避命令による対象物の指定とこの対象物に対する種類と範囲の指定に基づいて、定義情報を参照して指定された種類と、範囲に含まれる対象物を特定することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項3】 上記データベースシステムは、データベース処理言語を用いてアクセスでき、上記データベースアクセス手段は、データベース処理言語を用いて上記データベースシステムにアクセスすることを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項4】 上記制御手段は、保存媒体に保存する情報をデータベース処理言語が用いるコマンドの記述形式に変換して保存することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項5】 上記制御手段は、保存媒体に退避すべき対象物として定義情報を保存する場合に、保存する定義情報をデータベース処理言語による定義情報生成のための定義コマンドに変換して、保存媒体に保存することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項6】 上記制御手段は、保存媒体に退避すべき対象物としてデータを保存する場合に、保存するデータをデータベース処理言語によるデータ挿入のためのデータ挿入コマンドに変換して、保存媒体に保存することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項7】 上記制御手段は、保存媒体に退避すべき対象物の中に置換可能な変数がある場合、その変数の値を置換可能にする置換情報を生成し、これを保存媒体に保存することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項8】 上記データベースシステムの定義情報は、データベース処理言語による実行形式の処理手順を有しており、上記データベースシステムは、さらに、上記処理手順のソースコードを保持するとともに上記定義情報にある実行形式の処理手順を実行してデータを処理するプログラムを有しており、上記データベースシステム退避装置は、さらに上記プログラムをアクセスするプログラムアクセス手段を有しており、上記制御手段は、上記定義情報にある実行形式の処理手順が退避すべき対象物になった場合でもこれを退避すべき対象物とはせず、代わりに退避すべき処理手順のソースコードを有するプログラムを退避すべき対象物とし、上記プログラムアクセス手段を用いて退避すべきプログラムをアクセスして取り出し、保存媒体に保存することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項9】 上記制御手段は、データベースのデータ記憶領域を確保するためのデータベース処理言語による領域確保コマンドを生成して保存することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項10】 上記データベースシステムは、さらに、データベースへのアクセス履歴情報をデータの識別子とともに記録するジャーナルを有し、上記制御手段は、上記ジャーナルから識別子を検索することにより、保存媒体に退避すべき対象物に関連するデータのアクセス履歴情報を選択し、これを退避すべきジャーナルとして、保存媒体に保存することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項11】 上記退避手順生成手段は、退避すべき手順を生成する際に、退避すべき対象物間の依存関係を特定し、上記制御手段は、上記退避手順生成手段が特定した依存関係を依存情報として保存することを特徴とする請求項1記載のデータベースシステム退避装置。

【請求項12】 上記請求項1記載のデータベースシステム退避装置により保存媒体に保存された保存データを入力し、上記保存データに基づいてデータベースシステムを復元することを特徴とするデータベースシステム復元装置。

【請求項13】 上記請求項1記載のデータベースシステム退避装置により保存媒体に保存された保存データを入力し、上記保存データに基づいてデータベースシステムを新たに構築するデータベースシステム移行装置。

【請求項14】 上記請求項4、5、6記載のデータベースシステム退避装置により、データベース処理言語のコマンドの記述形式で保存媒体に保存された保存データを入力し、データベース処理言語により記述されたコマンドを実行することにより、データベースを復元又は移行することを特徴とするデータベースシステム復元装置又はデータベースシステム移行装置。

【請求項15】 上記請求項7記載のデータベースシステム退避装置により保存媒体に保存された保存データを入力し、置換可能な変数に新たな変数の値を与え、置換情報に基づいて置換可能な変数を新たな変数の値に置き換えながらデータベースシステムを新たに構築することを特徴とするデータベースシステム移行装置。

【請求項16】 上記請求項8記載のデータベースシステム退避装置により保存媒体に保存された保存データを入力し、データベースシステムを新たに構築するとともに、プログラムの実行時に、定義情報に実行形式の処理手順が存在しない場合、プログラムにある処理手順のソースコードをコンパイルして実行形式の処理手順を新たに作成することを特徴とするデータベースシステム移行装置。

【請求項17】 上記請求項11記載のデータベースシステム退避装置により保存媒体に保存された保存データを入力するとともに、復元仕様又は移行仕様を入力し、上記保存データの依存関係を参照することにより、復元又は移行するデータを復元仕様又は移行仕様に適合するように評価して、データベースを復元又は移行する復元手順生成手段又は移行手順生成手段を有することを特徴とするデータベースシステム復元装置又はデータベースシステム移行装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は、データベースシステムのプログラム及びデータの退避、復元、移行方式に関するものである。

【0002】

【従来の技術】 関係データベースが急速に普及しつつある。そこでは、データベースの処理言語としてSQL (JIS X3005 ISO/IEC 9075: Database Language SQL) と呼ばれる国際標準言語が用いられることが一般的である。この

ため、以下では関係データベースと国際標準データベース処理言語SQLを前提として説明する。

【0003】 業務プログラムとデータベースの基本構成を図15に示す。一般にデータベースシステムは、処理対象のデータそのものに加えて、データの内容を記述したデータベースの定義情報と、これら定義情報を参照しながら動作する業務プログラムとから構成される。これら定義情報はカタログ、リポジトリ、データディレクトリ/ディクショナリ等と呼ばれる。また、データに対しては、常時更新が行なわれおり、これら更新の履歴が特殊なファイルに記録される。更新の履歴を格納する特殊なファイルはジャーナル、ログなどと呼ばれる。

【0004】 一方、SQLなどのデータベース処理言語はデータベース処理（データベースからの所要データの検索、データベース中のデータの更新、削除、挿入、及びこれらデータ操作を行なうためのデータの定義等）を行なうことを目的とした言語であり、業務で必要となる通信、画面制御などはプログラミングが不可能である。このため、業務プログラムは、データベース処理の部分をSQLで記述し、その他の部分をCOBOLなどの従来の言語で記述して作成されることが一般的である。

【0005】 SQL文を含むプログラムをコンパイルすると、SQL以外の部分は通常のロードモジュールに、また、SQLの部分は、SQL手順にコンパイルされる。ここで、元SQL文のあった部分には、データベースシステムのデータベースシステム中核部を経由してコンパイルされたSQL手順を呼び出すための規定の手続呼びだし（図15ではsql exec）が替りにおかれ、またその引数としてデータベースシステムに登録されたSQL手順の識別子が与えられる。システムの実現方式にも依存するが、SQLを用いて作成されたロードモジュールとSQL手順は、SQLシステムの定義情報の一部として登録されることが一般的である。

【0006】 図15は業務プログラムとデータベースの基本構成を示す図であり、図においては、COBOL業務プログラムの中にSQLプログラムを作成した例を示している。このSQLプログラムはコンパイラによりコンパイルされる。コンパイラがSQLプログラムをコンパイルする場合には、SQLプログラムそれぞれに対して識別子を与える。この図においては、それぞれのSQLプログラムに対応するSQL手順に対して、No. 23とNo. 24が識別子として与えられた例を示している。COBOL業務プログラムは、コンパイルされるとロードモジュールという実行形式の形に変換されるが、このロードモジュール内にはSQL手順を実行するためのCALL文がおかれている。CALL文は、SQL手順の識別子23と24を引数にすることにより、カタログに登録されたSQL手順をコールする。こうしてSQL手順に記述されたデータベース処理を実行する。

【0007】 即ち、図16に示すように業務プログラム

を実行すると、通常のプログラムと同様に実行がなされるが、実行がデータベースシステムの処理要求の手続き呼びだし (sql exec) に達すると、この手続きからデータベースシステム中核部を経由して対応するSQL手順が取り出され、実行される。このようにして、業務プログラムが適宜データベースにアクセスしながら所要の業務を実現する。

【0008】次に、データベースの定義情報をについて説明する。データベースシステムは、データベースに関する定義情報を種々の方法で生成、管理している。例えばSQLでは、データベース情報をシステムで一元管理することにより、データ定義の重複の防止やデータベース内部での記憶情報の管理の効率化を実現している。これら定義情報の集まりをカタログと呼ぶ。SQLでは、これらシステムで管理される対象として以下のようなものがある。

(1) 実表。これは従来のファイルに相当するデータの本体である。実表を構成するフィールドを列と呼ぶ。列には各々データ型、長さなどの属性がある。

(2) ビュー。これは、従来の論理ファイルなどに相当し、実表に条件を付して検索した結果をあたかも実表のようにして見せるものである。即ち、データを、あるがままの状態以外に、一次加工、二次加工した形で見せることにより、データへの不要なアクセスの防止、共通したアクセスの一元化などを実現する。これは、複数の実表(ファイル)を結合して一つに見せる、といった複雑な場合も含まれる。また、(複数の)別のビューを参照することにより作成されたビューも存在する。実表とビューを総称して表と呼ぶ。

(3) スキーマ。これは従来のディレクトリなどに相当し、表等を区分けして整理するための入れ物を指す。表(実表とビュー)は、例えばスキーマ名をS1とし、表名をT1とするとS1.T1のようにスキーマ名を冠して名前付けされる。

(4) データベースの利用者。

(5) データベースの利用者が持っている権限。実表やビューに対する操作に対しては、これらを作成したユーザがすべての権限(選択、更新、削除、挿入)を有するが、これら権限を適宜他のユーザに与えることによって一定のユーザの集団の中でのデータベースへのアクセスを制御することができる。即ち、SQLではデータのアクセス権限を厳密に管理し、データの選択、更新、削除、挿入などの権限を実表、ビュー毎に、ユーザ毎に制御可能である。また、これらは権限は基本的に実表、ビューなどの生成者が保有するものであるが、これを権限を他人に付与したり、更にこの付与行為までを他人に認めることが可能である。付与行為までを認めるこをWITH GRANT OPTION付きで権限を付与する、と呼ぶ。

(6) データベースの業務プログラムとその中で使用さ

れるSQL手順。

【0009】SQLでは以上の(1)~(6)を例えれば以下のような情報としてカタログ中で管理する。

a) ユーザー一覧情報(ユーザ名、保有スキーマ数)

ユーザ名とそのユーザが保有するスキーマの数。

b) スキーマ一覧情報(スキーマ名、所有者名、格納表数)

スキーマの名前、スキーマを所有する所有者の名前、スキーマ中の表の数。

c) 表一覧情報(名前、列数、所属スキーマ名、ビューか実表か)

実表とビューを表として統一的に管理する。表の名前、表の列の数、表の所属するスキーマ名、実表かビューかの区別からなる。

d) 列一覧情報(名前、所属表名、所属スキーマ名、データ型、長さ)

列毎の情報。列の名前、その列が属する表の名前とその表が属するスキーマの名前、データ型、長さなど。

e) ビュー定義一覧情報(ビュー名、所属スキーマ名、定義本体)

各ビューに対して、その定義本体を管理する。ビューの名前とその属するスキーマ名、及びビュー定義の本体からなる。

f) ビュー対表の参照関係情報(ビュー名、所属スキーマ名、参照表名、参照表所属スキーマ名)

ビューは別の実表またはビューを参照することにより生成される。そのビューが参照しているこれら表毎にビュー名とそのビューが所属するスキーマ名、及びビューが参照している表の名前とその表が所属するスキーマ名がある。

g) 権限一覧情報(付与者、受領者、種類、対象、対象のスキーマ名、再付与可能性)

データベースの使用者に与えられている表に対するアクセスの権限(従来の、読とり可能、書き込み可能、実行可能、などの権限に相当)に関する情報からなる。例えば、権限を付与した人間、権限を受領した人間、権限の種類(当該表に対する選択権、更新権、挿入権、削除権)、権限の対象となる表、権限の付与行為まで与えているかどうか、からなる。特に、新たに表を作成した場合は、作成者はシステムから権限を受領したと見なされ、権限付与者には特別な値(SYSTEM)が設定される。また、ビューを生成する際には、選択権が少なくとも必要である。また、生成されたビューが一定の条件を満たさない(例えば複数の表を参照している、など)場合には、更新が不可能となるため、ビューに対する更新、削除、挿入の権限は生成されない。

h) プログラム一覧情報(業務プログラム識別子、OS内プログラム所在)

業務プログラムの一覧。識別子とプログラムが実際に存在するOS下の位置(ディレクトリ等)からなる。

i) 手順一覧情報（業務プログラム識別子、手順識別子、手順本体）

上記の業務プログラム中で用いているSQLコマンドに対応してコンパイルにより生成されたSQL手順の一覧。プログラム識別子、SQL手順の識別子、及び手順の本体の内容からなる。

j) 手順対表参照関係情報（業務プログラム識別子、手順識別子、スキーマ名、表名）

業務プログラムの各手順が各々参照している表を管理する。手順とそれが参照する表毎に一つの情報が存在する。

【0010】次に、データベースとカタログの具体例について説明する。例えば、図17のようなデータベースの内容は図18～図27のようなカタログ情報として管理される。図17に示したデータベースにおいては、鈴木と田中という2人のユーザが存在しており（図18参照）、ユーザ鈴木はスキーマS1を所有しており、ユーザ田中はスキーマS2、S3を所有している（図19参照）。また、スキーマS1は実表T1とビューV1を有しており、スキーマS2は実表T2とビューV2を有しており、スキーマS3はビューV3を有している（図20参照）。また、ビューV1はT1を参照しているビューである（図23参照）。その定義は、

```
SELECT * FROM T1 WHERE t1  
1 > 100
```

であり（図22参照）、更新が可能なビューである（図24参照）。ビューV2はT1とT2を参照しているビューである（図23参照）。その定義は、

```
SELECT t11, t12, t21 FROM T  
1, T2 WHERE t11 = t23
```

であり（図22参照）、複数の表を参照しているため、更新が不可能なビューである（図24参照）。また、このビューは他人（鈴木）が所有する表T1を参照しているため、予め鈴木から田中に関する選択権の付与がなければならない（図24参照）。ビューV3はV2を参照しているビューである（図23参照）。その定義は、

```
SELECT v21 FROM V2
```

であり（図22参照）、複数の表を参照しているため、更新が不可能なビューである（図24参照）。プログラムPROG1にはSQL文が二つあり、各々P1、P2なるSQL手順である（図25及び図26参照）。プログラムPROG2にはSQL文が一つあり、P3なるSQL手順である（図25及び図26参照）。SQL手順P1はV2を参照している（図27参照）。SQL手順P2はT2を参照している（図27参照）。SQL手順P3はV1とV3を参照している（図27参照）。鈴木は田中にT1に関する選択権（参照する権利）を再付与権つきで与えている（図24参照）。以上のように、図18から図27に示したカタログ情報は、前述したデータベースの定義情報の具体例を示すものである。

【0011】更に、SQLなどを用いたデータベースシステムは、以下のような特長を持つ。

(1) データベースに対して実行された更新、削除などの履歴をすべてジャーナルに記憶することにより、万一のデータベースの破壊に対してもデータベースを必ず回復できる。

(2) 更に、索引の追加、アクセス権限の変更など、データベースに変化が生じ、カタログ中に登録されているSQL手順が最適な手続きでなくなった場合には、SQL手順がCOBOL業務プログラムを通じて実行される際に当該SQL手順が動的に再コンパイルされ、変化に対応した最適な手続きを再度生成し、これをカタログに登録しなおして実行を再開する。

【0012】

【発明が解決しようとする課題】以上のような特長を実現する一方で、逆に以下のような問題点が生じている。

(1) あるシステムの上で開発した業務をパッケージ化しようとすると、他の開発中の業務などを含め、すべてのデータベースの定義情報が一つのカタログ内で一元管理されているため、その中の必要な部分だけをとりだしてパッケージにまとめることが困難である。例えば、実表は複数のビューに参照されており、またビューは複数のビューに参照されている。また、ビューを参照するには権限が必要であり、これら権限が定義されている状態を、ビューを生成するたびに忠実に再現しなければならない。例えば、上記の例では、ビューV2を生成するためには、表T1、表T2の作成が必要であり、更にこれら表の作成には、前もってスキーマS1、S2と各々のスキーマの所有者である鈴木、田中のユーザが生成されなければならない。また、これら表が生成された後、更に表T1に関する選択権を鈴木から田中に対して付与しておかなければ、田中はT1を参照することができず、ビューV2が生成できない。このように、定義情報は相互に複雑にからみあっており、また時間軸に沿って履歴的な復元が必要となる。現在は、

a) これを人間が必要な部分、必要な順序を手作業で認識し、取り出して、

b) 更に、手作業により移行先で注意深く再現を行なっている。

即ち、従来より行なわれているソフトウェアのパッケージの作成、普及が困難であり、対象システム各々において人手による複雑なシステムインストール、開発作業が必要となっていた。

【0013】(2) 更に、最近は、LAN（ローカルエリアネットワーク）を通じて多くのマシンが接続され、ソフトウェア開発もあるマシンで行なわれるとその成果がLANを通じて複数の別のマシンに配布される、といった形態が普及している。しかし、上記のような問題があり、データベースをアクセスする業務プログラムはこのような仕組みが整ってもこれを有効に活用して、ソフ

トウェア配布を行なうことが困難であった。

【0014】(3) 更に、最近は、ソフトウェア開発全体を管理するCASE(Computer Aided Software Engineering)が普及しつつあるが、CASEの中で開発作業の定義情報を管理するリポジトリは上記のカタログと同様な位置付けのものであり、これら開発作業環境の移行や複数グループによる開発の分散を行う場合にも上記(1)の問題が生じている。

【0015】(4) これらの定義情報を履歴に沿って移行先で再現するためにはデータベースシステム自身に従来の機能に加えて特殊な機能が必要になる可能性がある。例えば、各スキーマ中に表がいくつある、といった情報を移行しようとすると、履歴にカタログの内容をそっくり保管し、パッケージインストール時にこの部分を取り出し、移行先で、システムのカタログの該当部分を直接変更することが可能である。しかし、このためには、通常のデータベース機能以外のカタログ処理の特殊なインターフェースを持つ必要がある。また、このため、COBOL、SQLと行った標準言語で開発されたパッケージであっても、特殊なインターフェースがベンダの仕様に依存してしまい、異なるベンダ間ではシステムの移行ができない。

【0016】(5) また、従来のジャーナルはデータベース全体を通じてのデータ変更の履歴を記録しており、データベースの保管の際には、その復元作業のため常にデータベース全体を保管(バックアップ)する必要がある。データベース全体の内のごく一部の部分にのみ変更が加えられていたと判っていても、常に全体を保管するため、システムの保守運用時間が長くなっている。

【0017】(6) また、システムの移行に際しては、業務プログラムとカタログ中に登録されたSQL手順との対応関係を維持するための識別子の生成が困難となる。特に、業務プログラムを別のマシン、別のSQLシステムに移行した場合には、そのマシン上で既に稼働中の業務プログラム・SQL手順が使用している識別子と、移行により持ち込んだ識別子が衝突する場合があり、移行が困難である。また、移行先では移行したSQL手順がデータ量などの関係で必ずしも最適なSQL手順にコンパイルされた状態ではない場合がある。

【0018】この発明は以上のような問題点を解決するためになされたもので、データベースシステムのプログラム及びデータの退避、移行、復元が容易に行える装置を得ることを目的とする。

【0019】

【課題を解決するための手段】第1の発明に係るデータベースシステム退避装置は、退避命令とそのパラメータを入力するコマンド解釈手段と、コマンド解釈手段により解釈された退避命令とそのパラメータに基づいて退避すべき対象物を特定する制御手段と、この制御手段によ

り特定された退避すべき対象物の退避手順を生成する退避手順生成手段を有したものであり、以下の要素を有するものである。

(a) 上記データベースシステムをアクセスするデータベースアクセス手段、(b) 退避すべきデータを保存データとして保存媒体に保存する保存媒体アクセス手段、(c) 上記データベースシステムの退避命令を入力して解釈するコマンド解釈手段、(d) 上記データベースアクセス手段によりデータベースシステムの定義情報をアクセスすることにより、退避すべき対象物の退避すべき手順を生成する退避手順生成手段、(e) 上記コマンド解釈手段により解釈された退避命令に基づき、上記データベースシステムの中から退避すべき対象物を特定するとともに、上記退避手順生成手段を動作させ、退避手順生成手段により生成された退避すべき手順に従って、退避すべき対象物を上記データベースアクセス手段によりデータベースシステムから取り出し、上記保存媒体アクセス手段により保存媒体に保存する制御手段。

【0020】第2の発明に係るデータベースシステム退避装置は、退避命令のパラメータとして、退避すべきユーザ名、スキーマ名、表名、ビュー名、プログラム名、権限名のいずれかを入力すると共に、これら退避すべき対象物の種類の限定と範囲の指定がなされることを特徴とし、制御手段はこの対象物の指定と、この対象物に対する種類と範囲の指定に基づいて、定義情報から対象物を特定することを特徴とするものである。

【0021】第3の発明に係るデータベースシステム退避装置は、データベースシステムがSQL等のデータベース処理言語を用いてアクセスでき、データベースアクセス手段はSQL等のデータベース処理言語を用いて、データベースシステムをアクセスすることを特徴とするものである。

【0022】第4の発明に係るデータベースシステム退避装置は、保存媒体に情報を保存する場合に、この情報をSQL等のデータベース処理言語により記述して保存することを特徴とするものである。

【0023】第5の発明に係るデータベースシステム退避装置は、定義情報を退避する場合にSQL等のデータベース処理言語により、定義情報を生成するための定義コマンドに変換して退避することを特徴とするものである。

【0024】第6の発明に係るデータベースシステム退避装置は、データを退避する場合にデータをSQL等のデータベース処理言語によるデータ挿入のためのデータ挿入コマンドに変換して退避することを特徴とするものである。

【0025】第7の発明に係るデータベースシステム退避装置は、ユーザ名やファイル名等の置換可能な変数が退避すべき対象物に存在する場合に、これらユーザ名やファイル名等の置換可能な変数を置換情報として生成し

これを保存することを特徴とするものである。

【0026】第8の発明に係るデータベースシステム退避装置は、データを処理するプログラムはSQL等のデータベース処理言語による処理手順を呼び出して実行すると共にそのプログラムが他の機種に移植された場合のことを考慮して、予めデータベース処理言語による処理手順をソースコードとしてプログラム内に保持していることを特徴としており、定義情報内にある実行形式の処理手順が退避すべき対象物になった場合でも、この実行形式の処理手順を退避せずに、その処理手順のソースコードを有するプログラムを退避することを特徴とするものである。

【0027】第9の発明に係るデータベースシステム退避装置は、データベースのデータを退避せず単にデータベースのデータ記憶領域を確保するための領域確保コマンドを生成して保存することを特徴とするものである。

【0028】第10の発明に係るデータベースシステム退避装置は、データベースシステムがアクセス履歴情報を記録するジャーナルを有しており、アクセス履歴情報は、例えば表名等の識別子と共に記録されており、制御手段は退避すべき対象物に関連したアクセス履歴情報のみを表名などの識別子を元に検索し退避ようにしたものである。

【0029】第11の発明に係るデータベースシステム退避装置は、ユーザ、スキーマ、表、ビュー、権限、プログラム間の依存関係を特定する退避手順生成手段を有し、この退避手順生成手段が生成した依存関係を依存情報として保存する制御手段を有することを特徴とするものである。

【0030】第12の発明に係るデータベースシステム復元装置は、前述したデータベースシステム退避装置により退避された保存データを入力することにより、データベースシステムの一部または全部を復元する事を特徴とするものである。

【0031】第13の発明に係るデータベースシステム移行装置は、前述したようなデータベースシステム退避装置により退避された保存データを入力することにより、他の機種或いは他のマシン上に新たにデータベースシステムを構築することを特徴とするものである。

【0032】第14の発明に係るデータベースシステム復元/移行装置は、前述したデータベースシステム退避装置がSQL等のデータベース処理言語により記述されたコマンドの形式でデータを退避している場合に、このデータベース処理言語により記述されたコマンドを入力して実行することにより、データベースシステムを復元または移行することを特徴とするものである。

【0033】第15の発明に係るデータベースシステム移行装置は、前述したデータベースシステム退避装置において、保存したデータの中に置換可能な変数が存在することを示す置換情報が存在する場合に、その置換情報

に基づき置換可能な変数に新たな変数を与えて、変数の値を置き換ながらデータベースシステムを新たに構築することを特徴とするものである。

【0034】第16の発明に係るデータベースシステム移行装置は、前述したデータベースシステム退避装置が実行形式の処理手順を保存せず、かわりに処理手順のソースコードを有しているプログラムを保存している場合に、実行時に処理手順が定義情報に存在しないことを条件として、実行時に処理手順のソースコードをコンパイルして実行形式の処理手順を新たに定義情報に作成することを特徴とするものである。

【0035】第17の発明に係るデータベースシステム復元/移行装置は、前述したデータベースシステム退避装置により退避された保存データを入力するとともに、復元仕様又は移行仕様を入力し、上記保存データの依存関係を参照することにより、復元又は移行するデータを復元仕様又は移行仕様に適合するように評価して、データベースを復元又は移行する復元/移行手順生成手段を有することを特徴とするものである。

【0036】

【作用】第1の発明におけるデータベースシステム退避装置においては、制御手段が退避命令に指定されたパラメータから保存すべき対象物を特定すると共に、退避手順生成手段が定義情報を参照することにより、退避すべき手順を時間軸に沿って矛盾なく生成する。従って他の業務開発中、あるいは他の業務が稼働中であるデータベースシステムから、指定した業務に必要な定義情報を時間軸、相互の関係を保存したまま抽出できる。これを用いて業務動作に必要な情報を復元、移行することができる。

【0037】第2の発明におけるデータベースシステム退避装置は、退避命令としてユーザ名、スキーマ名、表名、ビュー名、プログラム名、権限名のいずれか1つが指定されると共にその種類と範囲が指定され、制御手段は退避命令による対象物と、その種類と範囲に基づいて、退避に必要な対象物を定義情報から判定して、退避すべき対象物として特定するので、コマンド解釈手段に入力する退避命令には必要最少限の情報を入力することにより、自動的に退避すべき対象物が特定されることが可能になる。

【0038】第3の発明に係るデータベースシステム退避装置は、データベースシステムがデータベース処理言語によりアクセスでき、データベースアクセス手段がデータベース処理言語を用いてデータベースシステムをアクセスするため、特別なデータベースアクセス手段を持つ必要が無く、従来のデータベースシステム及び従来のデータベースアクセス手段をそのまま用いてデータベースシステムを退避することが可能になる。

【0039】第4の発明におけるデータベースシステム退避装置は、データベースシステムの情報を退避する場

合に、SQL等のデータベース処理言語によるコマンドに変換して保存するため、これを入力してデータベースを復元する場合、或いは別なシステムにデータベースシステムを移行する場合に、特別な復元手段或いは、移行手段というものを開発する必要が無く、従来のデータベース処理言語を処理できる手段が備わっていれば、データの復元或いはデータの移行が可能になる。

【0040】第5の発明におけるデータベースシステム退避装置は、定義情報を退避する場合に、SQL等のデータベース処理言語により定義情報を生成するための定義コマンドに変換して保存するため、これを入力してデータベースを復元する場合、或いは別なシステムにデータベースシステムを移行する場合に、特別な復元手段或いは、移行手段というものを開発する必要が無く、従来のデータベース処理言語を処理できる手段が備わっていれば、データの復元或いはデータの移行が可能になる。

【0041】第6の発明におけるデータベースシステム退避装置は、第5の発明におけるデータベース退避装置が定義情報に対してデータベース処理言語による定義コマンドを用いたのに対して、データを退避する場合にはデータベース処理言語によるデータ挿入のためのデータ挿入コマンドに変換して退避するものである。従って、データを復元する場合、或いはデータを新たなシステムに移行する場合でも、従来からのデータベース処理言語を処理できる手段が備わっていれば、データを復元或いは移行することが可能になる。

【0042】第7の発明におけるデータベースシステム退避装置は、データベースシステムを他のシステムに移行するような場合に、移行先の他のシステムに既に同一ユーザ名や同一ファイル名等が存在する場合に、これらの名が重複することを避けることを可能にしたものであり、ユーザ名やファイル名等の置換可能な変数を予め置換情報として生成し、保存するものである。この置換情報を用いることにより、保存した情報を新たなシステムに移行する場合に、変数の値を、移行しようとして居るシステムに既に存在しているユーザ名やファイル名と重複しない名前に変えることが可能になる。

【0043】第8の発明におけるデータベースシステム退避装置は、定義情報にある実行形式の処理手順を退避することなく、またデータベースシステムにアクセスするプログラムに処理手順のソースコードをもたせておき、このプログラムを退避することを特徴としている。このように実行形式の処理手順を退避しないことにより、退避するデータ量を削減することが可能になる。また処理手順をソースコードで退避することにより、処理手順移行先のシステムの環境で再コンパイルすることが可能になり、移行先のシステム環境に最適な実行形式の処理手順を生成することができる。

【0044】第9の発明におけるデータベースシステム退避装置は、データベースのデータを退避することな

く、データベースのデータ記憶領域のみを確保するための領域確保コマンドを生成するため、退避すべきデータ量の削減が図れると共に、他のシステムにデータベースを移行する場合、定義情報が移行されると共に、データベースの領域のみが確保されるため、移行先においてデータを新規に蓄積していくような場合に、移行元のデータを移行せず領域のみを確保することが可能になる。

【0045】第10の発明におけるデータベースシステム退避装置は、従来のジャーナルを退避する場合にアクセス履歴情報をすべて退避していたのに対して、識別子により関連するアクセス履歴情報を選択できるようになる。従って制御手段が特定した退避すべき対象物に関連するアクセス履歴情報のみが退避可能になる。

【0046】第11の発明におけるデータベースシステム退避装置は退避手順生成手段がユーザ、スキーマ、表、ビュー、権限、プログラム間の依存関係を特定し、制御手段がこの依存関係を保存するため、この保存されたデータからデータを復元する場合、或いは移行する場合に依存情報を参照することにより、データの選択、或いは削除が可能になる。即ち、依存関係とは、データベースシステムの定義情報に示されている各データベースの構成要素の関連を抽出したものであり、この依存関係を参照するということは、即ちデータベースシステムの定義情報を参照するということに等しく、この依存関係を保存データ内に保存しておき、復元移行の際に、この依存関係を参照するということはデータベースシステムの定義情報を参照するに等しく、結果として、保存されたデータの中から更に別なデータを抽出したり、或いは余分なデータを削除する場合に矛盾なく抽出或いは削除することが可能になる。

【0047】第12の発明におけるデータベースシステム復元装置は、データベースシステム退避装置により保存されたデータを入力して、データベースシステムを復元するものであり、データベースシステム退避装置が保存媒体に対して、データを時間軸或いは相互の依存関係を矛盾なく保存しているため、この保存データを順に入力することにより、元のデータベースシステムを復元することができる。

【0048】第13の発明におけるデータベースシステム移行装置は、同様に保存媒体が時間軸或いは相互関係において、データを矛盾無く保存しているために、移行の際には保存されたデータを順に新たなシステムにインストールしていくことにより、必要な定義情報、或いは必要なデータを新たなシステム先に構築することが可能になる。

【0049】第14の発明におけるデータベースシステム復元/移行装置は、保存されたデータがSQL等のデータベース処理言語によるコマンドによって記述されているため、このコマンドを実行することにより、定義情報或いはデータベースのデータ等を復元または移行する

ことが可能になる。このようにデータベース処理言語により記述されたコマンドを実行するだけで、定義情報、或いはデータ等を復元、または移行することが可能なため、従来から存在するデータベース処理言語の処理手段をそのまま用いることにより、データベースシステムの復元移行が可能になる。従ってデータベースシステムを復元または移行するための特別な復元手段、或いは特別な移行手段というものが不要になる。

【0050】第15の発明におけるデータベースシステム移行装置は、データベースシステムを他のシステムに移行する場合であって、移行先のシステムに既に重複するユーザ名やファイル名等が存在する場合に、置換情報にある変数の値を置き換えることが可能になるため、移行先においてユーザ名やファイル名等が重複することなく、システムの移行をすることが可能になる。

【0051】第16の発明におけるデータベースシステム移行装置は、データベースシステムを新たなシステムに移行する場合に、実行形式の処理手順を保存データから移行するのではなく、保存されたプログラム内にあるソースコードをコンパイルすることにより、実行形式の処理手順を定義情報内に作成する。従って、移行先のシステム環境にあった最適な実行形式の処理手順が生成される。

【0052】第17の発明におけるデータベースシステム復元/移行装置は、復元仕様又は移行仕様を入力し、復元/移行手順生成手段が上記保存データの依存関係を参照することにより、復元又は移行するデータを復元仕様又は移行仕様に適合するように評価して、データベースを復元又は移行するので、保存データすべてを用いてシステムの復元、移行をするのではなく、保存データの一部を用いてシステムの復元、移行をすることが可能になる。

### 【0053】

#### 【実施例】

実施例1. 従来例で説明したように、データベースの処理言語として、SQLと呼ばれる国際標準言語が用いられることが一般的である。このため、以下では関係データベースと国際標準データベース処理言語SQLを前提として説明するが、内容的には他のデータベース、あるいは他のデータベース言語でも同様に適用できる。又、この実施例では、従来例で示した図17～図27に示したデータベースシステムを具体例として説明する。図1はこの発明の一実施例を示すデータベース及びプログラム退避/移行/復元装置の一例を示す図である。ここで退避とは、既に存在しているデータベース及びプログラムの一部又は全部を磁気テープや磁気ディスク等の保存媒体に複写することをいう。又移行とは、前述した退避により保存媒体に複写してあるデータベース及びプログラムを用いて、ある計算機システムにおいて動作しているデータベースシステムを他の計算機システムに移植す

ることを言う。又復元とは、前述した退避により保存媒体に複写してあるデータベース及びプログラムを元の計算機システムに複写し直すことをいう。ここで、移行と復元の違いは、移行が他の計算機システムに対して行われるのに対して、復元が元の計算機システムに対して行われる点である。

【0054】以下、図に基づいてデータベース及びプログラム退避/移行/復元装置の各部について説明する。

#### 1) コマンド解釈装置

ユーザへのインターフェースを提供し、ユーザからの退避、復元、移行などの要求とその対象物のリストを受取り、正当性を確認してこれら要求を制御部に送る。又、移行の場合は、必要に応じてユーザ名などの置換指示をユーザより受取り、これを制御部に送る。

#### 【0055】2) 制御部

装置全体を制御すると共に、データベース及びカタログアクセス装置を用いて、ユーザから指定された退避、復元、移行の対象物を特定する。

【0056】退避の場合には、退避移行復元手順生成装置を呼び出すことによりこれら特定された対象物と処理要求から必要な定義情報、置換情報、依存情報を生成してこれをSQLの実行形式にして保存媒体アクセス装置により保存媒体に順次記憶する。更に、指定によりデータ自身（実表の内容）をデータベース及びカタログアクセス装置により読みだし、これをSQLの実行形式に変換して保存媒体アクセス装置により保存媒体に記憶する。更に、指定によりジャーナルをデータベース及びカタログアクセス装置により読みだし、これを保存媒体アクセス装置により保存媒体に記憶する。更に、指定によりプログラムをプログラムアクセス装置により読みだし、これを保存媒体アクセス装置により保存媒体に記憶する。また、以上の動作の際、ビューを評価して実表とする指定がある場合、あるいはデータベースの実表の内容に対してその一部をSQLコマンドの実行により変形したい要求がある場合には、適宜ビューSQL実行装置によりこれを実行する。

【0057】復元移行の場合には、保存媒体から定義情報、置換情報、依存情報を読みだし、これら情報と指定された対象物から退避移行復元手順生成装置を呼び出すことにより復元移行対象を選択する。これら特定された対象物に対して置換が必要な場合、置換する内容をコマンド解釈装置を経由してユーザから入力する。次に指定により置換情報に従って必要な置換を行なながら、特定された対象物に関する定義情報を保存媒体から保存媒体アクセス装置により読みだし、データベース及びカタログアクセス装置により復元移行する。これら定義情報はSQLの定義系のコマンドの形となっているため、この操作はこれらコマンドを実行することによりなされる。更に、指定によりデータ自身（実表の内容）及びジャーナルを保存媒体アクセス装置により保存媒体から読みだ

し、これをデータベース及びカタログアクセス装置によりシステムに復元移行する。更に、指定によりプログラムを保存媒体アクセス装置により保存媒体から読みだし、これをプログラムアクセス装置によりデータベースシステムに記憶する。また、以上の動作の際、ビューを評価し、実表とする指定がある場合、あるいはデータベースの実表の内容に対してその一部をSQLコマンドの実行により変形したい要求がある場合には、適宜ビューSQL実行装置によりこれを実行する。

#### 3) 退避移行復元手順生成装置

制御部から対象物を受取り、これらに対して必要な定義情報と定義の順序を生成する。退避の場合には、これら生成に必要な情報はデータベース及びカタログアクセス装置からカタログをアクセスすることにより得られる。復元移行の場合には、これら生成に必要な情報は保存媒体中にある依存情報をアクセスすることにより得られる。

#### 4) ビューSQL実行装置

指定されたビュー、またはSQL文を実行し、その結果を制御部に送る。

#### 5) 保存媒体アクセス装置

保存媒体（磁気テープ、ディスク等）に対して指定した内容を読みだし、書き込みする装置。

#### 6) データベース及びカタログアクセス装置

データベースの内容及びカタログの内容を読みだし、書き込みする装置。この場合には、データベースシステムに対して必要なSQL文を実行することで機能が実現される。また、この装置によりジャーナルの内容の読みだし、またはジャーナルの内容をデータベースに適用し、データベースを最新の状態にする。これらジャーナル操作に対するSQLの規格は存在しないが、システム毎に何等かの手段が用意されている。例えば、ジャーナルの読みだし、設定は通常のOSのファイル読みだし、生成により実行し、ジャーナルの実行はRECOVERコマンドなどによりデータベースシステムが実行する。

#### 7) プログラムアクセス装置

業務プログラムを読みだし、設置する装置。通常のOSのファイルのアクセス機能が備わっていれば充分である。

【0058】次に動作について説明する。

#### 1) コマンド解釈装置の動作

ユーザからの退避移行復元要求はまずコマンド解釈装置により解釈され、内容の正当性が確認されて、これらを内部形式に変換し、制御部に送る。コマンド解釈装置は、以下のような指定を受け付ける。

動作指定 = BACKUP : 退避動作

RESTORE : 復元、移行先インストール動作

対象物のリスト = 対象物、対象物、...

対象物 = ユーザ名（ユーザの所有物全ての指定の省略形と解釈）

スキーマ名（スキーマ下の全ての表を指定したと解釈）

スキーマ名.表名（表を指定したと解釈）

権限名

プログラム名

対象種類指定 = D : データ

C : カタログ

P : プログラム

これらの組み合わせ、例えばD+P等

対象範囲指定 = ONLY : 指定された対象物のみを対象物とする。

DOWN : 指定された対象物とこれらが参照している対象物すべてを対象物とする。

ALL : 指定された対象物とこれらが参照している対象物すべてとこれらを参照している対象物すべてを対象物とする。

制限範囲指定 = ユーザ名リスト（これらユーザ以外のユーザのスキーマ、表等を対象としない）

スキーマ名リスト（これらスキーマ以外のスキーマ、表等を対象としない）

ビューの内、ビューを実行して実表としたいもののリスト。実表の内、それに対してデータの選択、変形を行うためのSQL文のリスト。

#### 【0059】2) 制御部の動作

制御部は、コマンド解釈装置からの要求を解釈し、退

避、移行、復元操作の全体を制御する。

【退避の場合】

(1) 制御部は、コマンド解釈装置から受け付けた対象物を退避移行復元手順生成部におくり、対象物の一覧と定義情報、置換情報、依存情報を受け取る。対象物の一

覧として以下のものがある。

U : ユーザー一覧

S : スキーマー一覧

T : 実表一覧

V : ビュー一覧

A : 権限一覧

P : プログラム一覧

R U : 制限ユーザー一覧

R S : 制限スキーマー一覧

(2) ビュー一覧V中のビューで、ビューを評価し、実表とする指定のあるものは、これを評価し、このビューを実表とする。定義情報、依存情報において、このビューに対応する情報を実表の情報に置き換える(CREATE VIEW文をCREATE TABLE文に置き換える)。また、この実表を記憶しておく。また、このビューが参照しているビュー及び実表のうち、このビューのみにより参照されているものを定義情報から除く。

(3) 対象種類指定にC(カタログ)があれば、置換情報、定義情報、依存情報を保存媒体アクセス装置により保存媒体に格納する。

(4) 対象種類指定にD(データ)があれば、実表一覧Tにある実表の内容をデータベース及びカタログアクセス装置によりとりだし、これをINSERTコマンドとして保存媒体アクセス装置により保存媒体に記憶する。保存媒体に記憶する実表がビュー評価により(2)で生成されたものであれば、(2)で記憶しておいた実表の内容を保存媒体に記憶する。また、保存媒体に記憶する実表に対してSQL文の実行の指定があればこれをビューSQL実行装置により実行し、その結果をINSERTコマンドの形にして保存媒体に記憶する。また、対象種類指定にD(データ)があれば、ジャーナルをデータベース及びカタログアクセス装置によりとりだし、これを保存媒体アクセス装置により保存媒体に記憶する。この際、ジャーナルには操作対象となった実表の識別子が付されており、実表一覧Tにある表に関するジャーナル内容のみを取り出す。対象種類指定にP(プログラム)があれば、P(プログラム)にあるプログラムをプログラムアクセス装置によりとりだし、これを保存媒体アクセス装置により保存媒体に記憶する。

【0060】[復元移行の場合]

(1) 保存媒体アクセス装置から定義情報、置換情報、依存情報を読み出す。

(2) 制御部は、コマンド解釈装置から受け付けた対象物と依存情報を退避移行復元手順生成部におくり、対象物の一覧を受け取る。対象物の一覧は、

U : ユーザー一覧

S : スキーマー一覧

T : 実表一覧

V : ビュー一覧

A : 権限一覧

P : プログラム一覧

R U : 制限ユーザー一覧

R S : 制限スキーマー一覧

(3) コマンド解釈装置により置換指定をユーザから受け取り、その置換内容で置換ファイルを置き換える。

(4) 対象種類指定にC(カタログ)があれば、置換情報により定義情報を置き換えながら、対象物に関する定義情報をデータベース及びカタログアクセス装置により復元、移行する。この場合は、SQLの定義コマンドCREATE USER文群、CREATE SCHEMA文群、CREATE TABLE、CREATE VIEW、GRANT文群を読みだし、置換情報の内容に従って置き換えを行なないながら、これら文を記憶している順に実行する。これら文は既に退避の際に時間的、空間的に矛盾のない順に記憶されているため、この順に実行を行なうだけよい。ビュー一覧V中のビューで、ビューを評価し、実表とする指定のあるものは、CREATE VIEWコマンドからCREATE TABLEに変換してこれを実行する。また、これをSQL文の一つとして記憶しておき、後に(5)で実行する。また、このビューが参照しているビュー及び実表のうち、このビューのみで参照されているものを定義情報から除く。これは依存情報中で、このビューのみに依存しているものをたどることにより識別できる。

【0061】(5) 対象種類指定にD(データ)があれば、実表一覧Tにある実表の内容を生成するINSERT文をデータベース及びカタログアクセス装置によりとりだし、これを実行してデータベースに復元する。また、この実表に対してSQL文の実行の指定があればこれをビューSQL文実行装置により実行し、その結果をT' として保存し、この表の内容を、  
DELETE \* FROM 表名  
で一旦廃棄し、

INSERT INTO

表名

SELECT \* FROM T'

ベースに記憶しておく必要はない。従って、その実表T' 1を削除しても構わぬことになる。この場合には、ビューV1をT' 1に対して評価した結果をT' として一時記憶しておき、実表T' 1が他のビューや他のプログラムにおいて参照されることが無いと判断された時点で実表T' 1を削除し、T' に記憶しておいたビューV1を実表として生成することが可能である。また、実表T' 1のデ

により先の結果で置き換える。この具体例について以下に説明する。例えば実表T 1が存在し、これを参照するビューV 1が存在するものとする。即ち、実表T 1に対してビューV 1がSQL文の実行により評価される。もし、実表T 1がビューV 1のみにより参照され、それ以外のビューには一切参照されないことが依存情報等を参照することにより判明した場合には、実表T 1をデータ

ータの内、例えば `t11=100` のデータのみを復元する場合などは、

```
SELECT * FROM T1  
WHERE t11=100
```

を直接実行し、その結果で T1 の元のデータを置きかえるものである。

(6) 対象種類指定に D (データ) があれば、保存媒体からジャーナルをとりだし、これをデータベースに適用する。この際、ジャーナルには操作対象となった実表の識別子が付されており、実表一覧 T にある表に関するジャーナル内容のみを取り出す。

(7) 対象種類指定に P (プログラム) があれば、P (プログラム) にあるプログラムをプログラムアクセス装置によりとりだし、これをデータベースに記憶する。

【0062】尚、データベースをあるマシンから別マシンへ移行する場合はこれら全ステップを実行する。またデータベースを同一マシン上で復元するためには

(5)、(6) の二つのステップを実行すればよい。これらは対象種類の指定で D (データ) のみを指定することにより実現される。また、指定により現在のデータ内容を破壊して復元する場合には、

```
DELETE * FROM 表名
```

によりデータベース内容を一旦破壊して復元を行う。

【0063】データベースの内容自身の移行には、IN SERT コマンドにより SQL コマンドとして運ぶのではなく、システムにより更に高速な形で保存することも可能である。例えば、データの内容をファイルの形で保存し、復元の時もファイルのまま直接置き換えを行う、などが可能である。この場合は上記の復元操作における DELETE 文などは不用となる。この後、業務が開始され、復元された業務プログラムを実行する。この際、SQL の手順一覧情報は移行しないため、移行先では SQL 手順にアクセスしようとしてカタログ中にこれが見つからない。このため、システムは当該 SQL 手順を再度コンパイルし、これを手順一覧情報に登録して実行を再開する。これにより、SQL システムの持つ動的再コンパイル機構を利用してことで、手順一覧情報も自動的に再現できることになる。

【0064】次にコマンド解釈装置が受け付ける指定と、その指定により制御部が退避移行復元手順生成装置から受け取る対象物に関する具体例について説明する。例えば、コマンド解釈装置に受け付けられる指定はいろいろな組み合わせが可能である。例えば、この指定した業務に必要な定義情報の取りだし方として以下のような場合が考えられる。

a) [必要な要素を全て指定する場合]

業務を構成しているユーザ、スキーマ、実表、ビュー、プログラム、権限を全て指定し、これによりシステムが指定されたこれら情報を、相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

b) [ユーザ名を与えて他はシステムが自動的に抽出する場合]

業務を構成する一つまたは複数のユーザ名のみを与えて、このユーザが所有するスキーマ、実表、ビュー、プログラム、権限を自動的に抽出し、これら情報を、相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

c) [スキーマ名を与えて他はシステムが自動的に抽出する場合]

業務を構成する一つまたは複数のスキーマ名のみを与えて、このスキーマを所有するユーザ、及びスキーマの中にある所有する実表、ビュー、権限を自動的に抽出し、これら情報を、相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

d) [表名を与えて他はシステムが自動的に抽出する場合]

業務を構成する一つまたは複数の表名のみを与えて、この表が属するスキーマ、及びそのスキーマを所有するユーザ、その表に関する権限を自動的に抽出し、これら情報を、相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

e) [ビュー名を与えて他はシステムが自動的に抽出する場合]

業務を構成する一つまたは複数のビュー名のみを与えて、このビューが属するスキーマ、及びそのスキーマを所有するユーザ、そのビューに関する権限、及びこのビューが参照しているビュー及び実表、これら参照に仮定されている権限を自動的に抽出し、これら情報を、相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

f) [プログラム名を与えて他はシステムが自動的に抽出する場合]

業務を構成する一つまたは複数のプログラム名のみを与えて、このプログラムが参照しているユーザ、スキーマ、実表、ビュー、権限を自動的に抽出し、これら情報を、相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

g) [権限名を与えて他はシステムが自動的に抽出する場合]

業務を構成する一つまたは複数の権限名のみを与えて、この権限及び権限を付与しているユーザ、権限が付与されているユーザ、権限対象の実表、ビューを自動的に抽出し、これら情報を、相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

h) 上記の a) ~ g) において、更に抽出された実表、ビューを参照しているビューの全体をも抽出し、これらビューの生成に必要なスキーマ、ユーザ、権限を抽出して、これらを含めて相互関係や履歴的実行順序を自動的に判断して必要な復元順序を生成する。

i) 上記の a) ~ g) 及びそれらと h) の組み合わせに

おいて、抽出された情報を参照しているプログラムの全体を抽出する。

【0065】次に従来例で示したデータベースの定義情報の例を用いてコマンド解釈装置に対して指定されたものに対して制御部が定義情報を参照して特定するユーザ一覧U、スキーマ一覧S、実表一覧T、ビュー一覧V、権限一覧A、プログラム一覧Pの値をいくつかの例で示す。以下に述べる例は前述したコマンド解釈装置が受け付ける指定の例として述べた(a)から(i)には直接対応していないが、その中の1つ、或いは複数を組み合わせて指定した場合を示している。また以下の例においては、説明を簡単にするためにコマンド解釈装置が受け付ける全ての指定を示しているものではなく、単に対象物のリスト、対象範囲、制限範囲指定の3つの指定のみを例示している。また、ユーザ一覧U、スキーマ一覧S、ビュー一覧V、実表一覧T、権限一覧A、プログラム一覧P、制限ユーザー一覧RU、制限スキーマ一覧RSは制御部が退避移行復元手順生成装置から受け取る値をそれぞれ示すものである。退避移行復元手順生成装置は、コマンド解釈装置に入力された指定から、これらユーザ一覧U、スキーマ一覧S、実表一覧T、ビュー一覧V、権限一覧A、プログラム一覧P、制限ユーザー一覧RU、制限スキーマ一覧RSを求め制御部に出力するが、その手順については後述する。また、例中に示すのの記号は全体の意味を示すものであり、特に指定がされないことを示すものである。

### 【0066】

例1 指定した実表を退避

対象物のリスト=鈴木、S1、  
S1. T1

対象範囲指定 =ONLY

制限範囲指定 =なし

ユーザ一覧U	=鈴木
スキーマ一覧S	=S1
ビュー一覧V	=φ
実表一覧T	=T1
権限一覧A	=φ
プログラム一覧P	=φ
制限ユーザー一覧RU	=φ
制限スキーマ一覧RS	=φ

例4 複数のビューと参照している実表を退避

対象物のリスト=鈴木、田中、S1、S2、  
S1. V1、S2. V2、  
S1. T1、S2. T2

対象範囲指定 =ONLY

制限範囲指定 =なし

ユーザ一覧U	=鈴木、田中
スキーマ一覧S	=S1、S2
ビュー一覧V	=V1、V2

### 【0067】

例2 複数の指定した実表を退避

対象物のリスト=鈴木、田中、S1、S2、  
S1. T1、S2. T2

対象範囲指定 =ONLY

制限範囲指定 =なし

ユーザ一覧U	=鈴木、田中
スキーマ一覧S	=S1、S2
ビュー一覧V	=φ
実表一覧T	=T1、T2
権限一覧A	=φ
プログラム一覧P	=φ
制限ユーザー一覧RU	=φ
制限スキーマ一覧RS	=φ

### 【0068】

例3 ビューとその参照している実表を退避

対象物のリスト=鈴木、S1  
S1. V1、S1. T1

対象範囲指定 =ONLY

制限範囲指定 =なし

ユーザ一覧U	=鈴木
スキーマ一覧S	=S1
ビュー一覧V	=V1
実表一覧T	=T1
権限一覧A	=φ
プログラム一覧P	=φ
制限ユーザー一覧RU	=φ
制限スキーマ一覧RS	=φ

### 【0069】

権限一覧A	=T 1 に関する田中の選択権
実表一覧T	=T 1, T 2
プログラム一覧P	= $\emptyset$
制限ユーザー一覧R U	= $\emptyset$
制限スキーマ一覧R S	= $\emptyset$

【0070】例3に対し、ビュー名のみを指定し、ビューが参照している表名を自動的に取り出すことを指定す

一が参照している表名を自動的に取り出すことを指定するのが例6である。

るのが例5である。

例5 ビューを退避、関連実表も自動退避

対象物のリスト	=S 1, V 1
対象範囲指定	=DOWN
制限範囲指定	=なし
ユーザー一覧U	=鈴木
スキーマ一覧S	=S 1
ビュー一覧V	=V 1
実表一覧T	=T 1
権限一覧A	= $\emptyset$
プログラム一覧P	= $\emptyset$
制限ユーザー一覧R U	= $\emptyset$
制限スキーマ一覧R S	= $\emptyset$

【0071】例4に対し、ビュー名のみを指定し、ビュ

例6 複数ビューを退避、関連実表自動退避

対象物のリスト	=S 1, V 1, S 2, V 2
対象範囲指定	=DOWN
制限範囲指定	=なし
ユーザー一覧U	=鈴木、田中
スキーマ一覧S	=S 1, S 2
ビュー一覧V	=V 1, V 2
実表一覧T	=T 1, T 2
権限一覧A	=T 1 に関する田中の選択権
プログラム一覧P	= $\emptyset$
制限ユーザー一覧R U	= $\emptyset$
制限スキーマ一覧R S	= $\emptyset$

【0072】

例7 ビューを退避、参照している表を退避

対象物のリスト	=S 1, V 2
対象範囲指定	=DOWN
制限範囲指定	=なし
ユーザー一覧U	=鈴木、田中
スキーマ一覧S	=S 1, S 2
ビュー一覧V	=V 2
実表一覧T	=T 1, T 2
権限一覧A	=T 1 に関する田中の選択権
プログラム一覧P	= $\emptyset$
制限ユーザー一覧R U	= $\emptyset$
制限スキーマ一覧R S	= $\emptyset$

【0073】

例8 ビューを退避、関連表を全て自動退避

対象物のリスト	=S 1, V 2
対象範囲指定	=ALL

制限範囲指定 =なし  
 ユーザー一覧U =鈴木、田中  
 スキーマ一覧S =S 1、S 2、S 3  
 ビュー一覧V =V 2、V 3  
 実表一覧T =T 1、T 2  
 権限一覧A =T 1に関する田中の選択権  
 プログラム一覧P = $\phi$   
 制限ユーザー一覧R U = $\phi$   
 制限スキーマ一覧R S = $\phi$

#### 【0074】

例9 関連の複雑なビューを退避  
 対象物のリスト=V 3  
 対象範囲指定 =DOWN  
 制限範囲指定 =なし  
 ユーザー一覧U =鈴木、田中  
 スキーマ一覧S =S 1、S 2、S 3  
 ビュー一覧V =V 2、V 3  
 実表一覧T =T 1、T 2  
 権限一覧A =T 1に関する田中の選択権  
 プログラム一覧P = $\phi$   
 制限ユーザー一覧R U = $\phi$   
 制限スキーマ一覧R S = $\phi$

#### 【0075】

例10 ユーザーの所有物を全て退避  
 対象物のリスト=鈴木  
 対象範囲指定 =DOWN  
 制限範囲指定 =なし  
 ユーザー一覧U =鈴木  
 スキーマ一覧S =S 1  
 ビュー一覧V =V 1  
 実表一覧T =T 1  
 権限一覧A = $\phi$   
 プログラム一覧P = $\phi$   
 制限ユーザー一覧R U = $\phi$   
 制限スキーマ一覧R S = $\phi$

#### 【0076】

例11 あるスキーマの下を全て退避  
 対象物のリスト=S 1  
 対象範囲指定 =DOWN  
 制限範囲指定 =なし  
 ユーザー一覧U =鈴木  
 スキーマ一覧S =S 1  
 ビュー一覧V =V 1  
 実表一覧T =T 1  
 権限一覧A = $\phi$   
 プログラム一覧P = $\phi$   
 制限ユーザー一覧R U = $\phi$   
 制限スキーマ一覧R S = $\phi$

#### 【0077】

例12 スキーマ下全てと表を退避  
 対象物のリスト=S 1、S 2、T 2  
 対象範囲指定 =DOWN  
 制限範囲指定 =なし  
 ユーザー一覧U =鈴木、田中  
 スキーマ一覧S =S 1、S 2  
 ビュー一覧V =V 1  
 実表一覧T =T 1、T 2  
 権限一覧A = $\phi$   
 プログラム一覧P = $\phi$   
 制限ユーザー一覧R U = $\phi$   
 制限スキーマ一覧R S = $\phi$

#### 【0078】

### 例13 プログラムと関連物を退避

対象物のリスト=PROG2  
対象範囲指定=DOWN  
制限範囲指定=なし  
ユーザー一覧U =鈴木、田中  
スキーマ一覧S =S1、S2、S3  
ビュー一覧V =V1、V2、V3  
実表一覧T =T1、T2  
権限一覧A =T1に関する田中の選択権  
プログラム一覧P =PROG2  
制限ユーザー一覧RU = $\phi$   
制限スキーマ一覧RS = $\phi$

### 【0079】

### 例14 プログラムと表を退避

対象物のリスト=PROG1、T1  
対象範囲指定=DOWN  
制限範囲指定=なし  
ユーザー一覧U =鈴木、田中  
スキーマ一覧S =S1、S2  
ビュー一覧V =V1、V2  
実表一覧T =T1、T2  
権限一覧A =T1に関する田中の選択権  
プログラム一覧P =PROG1  
制限ユーザー一覧RU = $\phi$   
制限スキーマ一覧RS = $\phi$

### 【0080】

#### 例15 スキーマ制限付でプログラムを退避

対象物のリスト=PROG2  
対象範囲指定=DOWN  
制限範囲指定=S1  
ユーザー一覧U =鈴木  
スキーマ一覧S =S1  
ビュー一覧V =V1  
実表一覧T =T1  
権限一覧A = $\phi$   
プログラム一覧P =PROG2  
制限ユーザー一覧RU = $\phi$   
制限スキーマ一覧RS = $\phi$

### 【0081】

#### 例16 ユーザ制限付でビューを退避

対象物のリスト=V3  
対象範囲指定=DOWN  
制限範囲指定=鈴木  
ユーザー一覧U =鈴木  
スキーマ一覧S =S1  
ビュー一覧V =V1  
実表一覧T =T1  
権限一覧A = $\phi$   
プログラム一覧P =PROG1  
制限ユーザー一覧RU = $\phi$   
制限スキーマ一覧RS = $\phi$

【0082】次に、図2に保存媒体の内容の例を示す。図2において、50は置換情報、60は定義情報、80は依存情報、100はデータ本体、110はジャーナル、120はプログラム群である。

【0083】置換情報50にはユーザ名の置換情報が記録されている。この例では、置換情報各々は\$ではじまる仮の名前で表されている。この例においては、ユーザ1が鈴木であり、ユーザ2が田中であることを示しており、この置換情報を置換しなければ、ユーザ1は鈴木、ユーザ2は田中として復元、或いは移行できる。しかし、移行先のシステムに既にユーザ名鈴木、又はユーザ名田中が存在している場合にはこの置換情報を用いることにより、例えばユーザ1を鈴木ではなく山田というユ

ーザ名に変更することが可能になる。すなわち、以後に出てくる\$U1という変数は全て山田に置き換えられてインストールされることになる。

【0084】次に、定義情報60は定義情報を生成するための情報がSQL文により、記述されている。61においてはユーザ1が生成され、62においてはユーザ2が生成されることを示している。63においては、次の64のスキーマ作成者をユーザ1に設定する。そして64においては、63において設定されたユーザ1の所有物としてスキーマS1を生成する。同様に、65から67においてはユーザ2に対してスキーマS2、S3が生成される。また68においては以下のSQL文の実行者をユーザ1に設定する。69においては実表T1が生成される。70においては実表T1を参照する権限をユーザ2に与えると共に、ユーザ2はその参照権限を他のユーザに与えるオプションを持っていることを宣言している。71においては実表T1から列t11が100以上のものだけをセレクトしてビューV1を生成することを定義している。次に、72～75においてはユーザ2に対してテーブル2を生成し、実表T1と実表T2からビューV2を生成し、更に、ビューV2からビューV3を生成することを記述している。

【0085】次に、依存情報80は、退避したデータベースシステムの定義情報から抽出された依存関係を示しているものである。この例においては81、82においてユーザ1及びユーザ2は何にも依存しないことを示している。また83においてはスキーマS1はユーザ1の存在が必要であることを示している。84と85においてはスキーマS2、S3はユーザ2の存在が必要であることを示している。即ち、スキーマ1を生成する場合には、ユーザ1が先に存在していなければならぬ。同様に、スキーマS2、S3を生成する場合にはユーザ2が存在しなければ生成できないことを示している。次に、86においては実表T1はスキーマS1の存在が必要であることを示している。87においては実表T2はスキーマS2の存在が必要であることを示している。88においてはビューV1は実表T1とスキーマS1の存在が必要であることを示している。89においてはビューV2が実表T1、T2及び、スキーマS2の存在が必要であるとともに、ユーザ1からユーザ2に対する実表T1に関する選択権が必要であることを示している。90においてはビューV3はビューV2とスキーマS3の存在が必要であることを示している。次に、89で示したユーザ1からユーザ2への実表T1に関する選択権は、ユーザ1とユーザ2と実表T1が存在することが必要であることを示している。92においてはプログラムP1はビューV2の存在が必要であることを示している。また、93においてはプログラムP2は実表T2が存在が必要であることを示している。94においては、プログラムP3はビューV1とビューV3の存在が必要である

ことを示している。

【0086】これらの依存情報は、従来例で説明した定義情報の一覧情報から抽出することが可能である。例えば、81、82はユーザー一覧情報を参照することにより抽出できる。また83～85はスキーマ一覧情報を参照することにより抽出することが可能である。また86、87は表一覧情報を参照することにより抽出可能である。また88、89は表一覧情報と、ビュー定義一覧及びビュー対表の参照関係及び権限一覧を参照する事により、抽出可能である。また、91は権限一覧を参照する事により抽出可能である。また92～94は手順対表参照関係から抽出することが可能である。

【0087】データ本体100はデータベースの実際のデータをSQL文のインサートコマンドにより記述したものであり、101は実表T1のデータ挿入コマンドであり、102は実表T2のデータ挿入コマンドである。

【0088】次にジャーナル110にはジャーナルの通常のファイルの形式の履歴データが存在するが、ここでは前述した退避すべき対象物に関する履歴情報が、表名という識別子により抽出されて記述される。例えば、この例においては、実表T1とT2が退避されるため、ジャーナル110には実表T1とT2という識別子がついた履歴データのみが、ここに付加されることになる。

【0089】次にプログラム群120には通常のファイル形式のプログラムが付加されるが、例えばこのプログラムをプログラムA、B、C、Dの4つのロードモジュールであるとするならば、このプログラムはCOBOL等の一般的のプログラム処理言語で記載されており、そのプログラム言語の一部にSQLで記述された手順がソースコードの形式で存在すると共に、その実行形式の手順を呼ぶためのコール文が存在している。この例においては、手順としてP1、P2、P3が退避べき対象物として特定されるが、手順P1、P2、P3は定義情報内にある実行形式の手順が保存されるのではなくCOBOLプログラムのロードモジュールA、B、C、D内にある手順P1、P2、P3のソースコードがプログラム群の一部として保存される。定義情報内にある実行形式の手順P1、P2、P3の保存はされない。

【0090】このように、通常プログラムに対応するロードモジュール中にデータベース処理言語の原プログラムを保持し、移動先のデータベース処理システム上でこのロードモジュールを実行し、データベース処理手順の識別子を用いて、移動先のデータベース処理システム上のデータベース処理手順を登録するカタログ領域にあるデータベース処理手順を検索し、これが存在しないことにより、ロードモジュール中のデータベース処理言語のプログラムを動的に再コンパイルし、これを移動先のシステムのデータベース処理手順を登録するカタログ領域に移動先のロードモジュールのファイル識別子と、データベース処理手順相対番号により新たに登録し、再び該

データベース処理部の実行を継続する。この際、データベース処理手順の識別子は、カウンタなどの値ではなく、当該アプリケーションの走行しているシステム上で実行時に採取したロードモジュールのファイル識別子を用いる。従って、このような移植動作によってもデータベース処理手順の一意性が保証されることになる。即ち、埋め込まれたデータベース処理言語により記述されたデータベース処理部には、データベース処理アプリケーション実行時に得た当該アプリケーションプログラムの通常プログラム部に対応するロードモジュールのオペレーティングシステム上のファイルの識別子と、当該データベース処理部の当該アプリケーションプログラム中での相対番号の値の組からなる識別子を与え、実行時に通常プログラムがデータベース処理が必要と成った場合には、実行時に得た自ロードモジュールのオペレーティングシステムのファイル識別子と、必要とするデータベース処理部のアプリケーション内相対番号を組として、データベース処理手順を登録したカタログ領域を検索し、これにより通常プログラム部から登録された所要のデータベース処理部を呼出す。

【0091】3) 退避移行復元手順生成装置の動作  
まず、渡された対象物指定を以下のように解釈し、対象

物の一覧を得る。尚、以下で(\*)の部分は、退避の場合にはデータベースのカタログ情報から識別が可能であり、また復元の場合には依存情報から識別が可能である。

【0092】(1) ユーザ一覧  $U = \emptyset$  : 指定されたユーザ全体を記憶する内部変数  
スキーマ一覧  $S = \emptyset$  : 指定されたスキーマ全体を記憶する内部変数  
ビュー一覧  $V = \emptyset$  : 指定されたビュー全体を記憶する内部変数  
実表一覧  $T = \emptyset$  : 指定された実表全体を記憶する内部変数  
権限一覧  $A = \emptyset$  : 指定された権限全体を記憶する内部変数  
プログラム一覧  $P = \emptyset$  : 指定されたプログラム全体を記憶する内部変数

【0093】(2) 指定された各対象物に対して指定されているものが以下のような場合、それぞれの対象物を次のように決定する。  
なお、以下に用いた記号 $\cup$ はその記号左側に記載した集合に、その記号右側に記載したものを合併する事を示す記号である。

a) ユーザ名が指定されている場合

ユーザー一覧  $U = \text{ユーザー一覧 } U \cup \text{このユーザ名}.$   
スキーマ一覧  $S = \text{スキーマ一覧 } S \cup \text{当該ユーザが所有している全てのスキーマ}$   
(\*)  
実表一覧  $T = \text{実表一覧 } T \cup \text{これらスキーマ中の全ての実表}$   
(\*)  
ビュー一覧  $V = \text{ビュー一覧 } V \cup \text{これらスキーマ中の全てのビュー}$   
(\*)

b) スキーマ名が指定されている場合

ユーザー一覧  $U = \text{ユーザー一覧 } U \cup \text{当該スキーマを所有するユーザ}$   
(\*)  
スキーマ一覧  $S = \text{スキーマ一覧 } S \cup \text{当該スキーマ}$   
実表一覧  $T = \text{実表一覧 } T \cup \text{当該スキーマ中の全ての実表}$   
(\*)  
ビュー一覧  $V = \text{ビュー一覧 } V \cup \text{当該スキーマ中の全てのビュー}$   
(\*)

c) スキーマ名、表名が指定されている場合

スキーマ一覧  $S = \text{スキーマ一覧 } S \cup \text{この表が所属するスキーマ名}$   
(\*)  
とする。更にこれらをこの表が実表であれば  
実表一覧  $T = \text{実表一覧 } T \cup \text{この表}$   
(\*)  
そうでなければ  
ビュー一覧  $V = \text{ビュー一覧 } V \cup \text{この表}$

d) 権限が指定されている場合

ユーザー一覧  $U = \text{ユーザー一覧 } U \cup \text{当該権限を付与しているユーザ} \cup \text{当該権限を受領しているユーザ}$   
(\*)  
当該権限の対象物に対し、上記c)を実行する。  
権限一覧  $A = \text{権限一覧 } A \cup \text{当該権限}$

e) プログラム名が指定されている場合

当該プログラムが参照している表に対し、上記c) を実行する。  
プログラム一覧P=プログラム一覧P ∪ 当該プログラム

上記の(\*)について、カタログからこれを得る場合については後述のデータベース及びカタログアクセス装置の説明の際に述べる。また、依存情報からこれを得る場合には、以下のようになる。

例1：あるスキーマに属する表一覧を得る。

=このスキーマに依存するビュー及び実表を取り出す。  
図2に示す例では、S1に属する表一覧は86と88からS1に依存するT1とV1とわかる。

例2：あるビューが参照している表一覧

=このビューが依存するビュー及び実表を取り出す。  
図2に示す例では、ビューV2が参照する表は89のV2の依存部を見て、T1、T2とわかる。

【0094】(3) 対象範囲指定がONLYとDOWNとALLの場合の対象物の特定は、以下のようにして行う。

a) ONLYの場合、指定された対象物に対して、以下の確認を行なう。

- ・実表一覧T、ビュー一覧Vにある実表、ビューが所属するスキーマがスキーマ一覧S中にあるか？
- ・スキーマ一覧S中のスキーマの所有者がユーザー一覧Uにあるか？
- ・ビュー一覧Vの参照しているビュー、実表がビュー一覧V、実表一覧Tにあるか？
- ・ビュー一覧Vの生成に必要な権限が権限一覧Aにあるか？

以上の条件を満たさない場合は、エラーとする。

b) DOWNの場合は、以下のようにして対象物の特定を行う。これは、以下の退避復元移行手順生成装置のうごきと同じであるが、説明のため、ここに記述する。

・現在のビュー一覧Vの各ビューvについて、以下の特定を行う。

ビューvが参照している各表tについて、

表tがビューの場合：ビュー一覧V=ビュー一覧V ∪ 表t

表tをビューvとしてこれを再帰的に繰り返す。

表tが実表の場合：実表一覧T=実表一覧T ∪ 表t

- ・ビュー一覧V、実表一覧Tが所属するスキーマでまだスキーマ一覧Sにないものをスキーマ一覧Sに加える。
- ・スキーマ一覧Sにあるスキーマでまだその所有者がユーザー一覧Uにないものをユーザー一覧Uに加える。

これにより、全ての対象物を指定せども、a) ONLYの場合の条件が自動的に満たされ、対象物の範囲は閉じたものとなる。

例1：ユーザー一覧U=田中、スキーマ一覧S=S2、ビュー一覧V=V2、実表一覧T=∅の場合、

V2が参照している実表T1、T2を実表一覧Tに加える。スキーマ一覧SにS1を、ユーザー一覧Uに鈴木を加

える。

例2：ユーザー一覧U=田中、スキーマ一覧S=S3、ビュー一覧V=V3、実表一覧T=∅の場合、V3が参照しているビューV2をビュー一覧Vに加える。スキーマ一覧SにS2を加える。更にV2が参照している実表T1、T2を実表一覧Tに加える。スキーマ一覧SにS1を、ユーザー一覧Uに鈴木を加える。

c) ALLの場合は、以下のようにして対象物の特定を行う。

- ・DOWNの場合の手順を実行する。
- ・B=実表一覧T
- ・D=∅
- ・L：現在のBの各表tについて
  - a) もし、全てのtについてこれを参照しているビューがなければ終わり
  - b) そうでなければtを参照している各ビューvについて
    - ・権限一覧A=権限一覧A ∪ tに関してvの所有者に出されている権限
    - ・ビュー一覧V=ビュー一覧V ∪ v
    - ・vが参照している表が全てB中にある場合：  
B=B ∪ v
    - ・D=D ∪ v
    - ・B=D、D=∅
    - ・Lに戻る。

これらDOWNとALLの場合を図示すると図3のようになる。

【0095】(4) 制限範囲指定で指定されたユーザ名群を制限ユーザー一覧RUに入れる（なければ制限ユーザー一覧RUは∅）。制限範囲指定で指定されたスキーマ名群を制限スキーマ一覧RSに入れる（なければ制限スキーマ一覧RSは∅）。

【0096】(5) ユーザー一覧U=ユーザー一覧U ∪ 制限ユーザー一覧RU（ユーザー一覧Uと制限ユーザー一覧RUの交わり）

スキーマ一覧S=スキーマ一覧S ∪ 制限スキーマ一覧RS（スキーマ一覧Sと制限スキーマ一覧RSの交わり）

【0097】(6) 以上で定まった実表一覧T、ビュー一覧Vからスキーマ一覧Sにあるスキーマ中にない実表、ビューを除く。

【0098】(7) 復元、移行の場合には、これらユーザー一覧U、スキーマ一覧S、ビュー一覧V、実表一覧T、権限一覧A、制限ユーザー一覧RU、制限スキーマ一覧RSを、指定されたパラメータとともに制御部に渡す。

【0099】(8) 退避の場合には以下の手順を実行する。

- 1) ユーザ一覧Uで指定されたユーザを生成するCREATE USER文を生成し、定義情報とする。この際、ユーザ名は個々に異なる置き換え可能な変数とし、この変数名とユーザ名自身を組として置換情報として順に記憶する。
- 2) 指定されたユーザ一覧U及び指定されたスキーマ一覧Sを基に、これらスキーマを生成するCREATE SCHEMA文を生成し、定義情報として記憶する。
- 3) これらユーザ、スキーマに関連する実表、ビュー、アクセス権を生成、定義するCREATE TABLE、CREATE VIEW、GRANT文を生成し、定義情報として記憶する（詳細は後述する）。
- 4) 上記において判明している依存関係を依存情報として記憶する。依存関係は以下のようにして識別される。
  - ・ユーザ一覧U中のユーザ名は何にも依存しない。
  - ・スキーマ一覧S中のスキーマ名はその所有者のユーザに依存するとして依存情報を生成する。
  - ・実表一覧T中の実表はそれが存在するスキーマに依存するとして依存情報を生成する。
  - ・ビュー一覧V中のビューは、ビュー一覧Vが参照している実表またはビューとそれに関する権限、及びビュー一覧Vが存在するスキーマに依存するとして依存情報を生成する。
  - ・権限は、権限付与者、権限受領者、権限対象の表に依存するとして依存情報を生成する。
  - ・プログラムはプログラムが参照している表に依存するとして依存情報を生成する。

【0100】次に退避復元手順生成装置が行う実表、ビュー、権限の保存の手順について説明する。実表、ビューは相互に関連があり、またビューを生成する際には権限が必要となる。従って、ビュー生成の文を実行する前にはビューが参照している表生成のCREATE TABLE文とビューが仮定している権限を定義するGRANT文が実行されてなければならない。従って、実表、ビュー、権限の保存の大まかな手順は以下のようになる。

(1) まず、該当する実表をすべて生成する。この際に各実表を生成する度に表に関して付与されている権限を定義する。

(2) 次に、これら実表のみを参照しているビューを生成する。この際に各ビューを生成する度にビューに関して付与されている権限を定義する。

(3) 次に、これら実表、及びビューのみを参照しているビューを生成する。この際に各ビューを生成する度にビューに関して付与されている権限を定義する。

(4) 現在までに生成された実表及びビューを参照しているビューがなくなると停止する。

【0101】以上の過程において、権限に対するGRANT生成の際、複数のGRANT文が生成される場合がある。例えば、表T1を伏見が作成し、この権限をWI

THGRANT OPTION付きで安藤に渡す。安藤は岩崎に更に渡す。この場合、伏見→安藤、安藤→岩崎の順にGRANT文を生成しないと正しく実行できない。この順序を正しく生成する手順が必要となる。以下に、もっとも一般的な形として上記手順により実表一覧が得られ、かつ対象範囲指定でALLが選択された場合に、CREATE TABLE文、CREATE VIEW文、GRANT文が正しい順で生成される手順の一例を示す。尚、CREATE USER文、CREATE SCHEMA文は各々ユーザ一覧で、スキーマ一覧Sからそのまま生成できる為、その説明は省略する。

【入力】

実表一覧T : 対象となる表一覧

【出力】

CREATE TABLE, CREATE VIEW, GRANT文の列、

【手順】

(1) ハッシュ表Bを空にする。Bは現在までにCREATE文を生成したすべての表、ビューを記憶する。

(2) ハッシュ表Dを空にする。Dは、現在までにCREATE文を生成したビュー、実表の内、まだ上位のビューで参照されている可能性のあるもの全体を示す。

(3) 実表一覧T中の各表tに対し、tがスキーマ一覧Sのいづれの要素のスキーマにもなければ何もしない。そうでなければ実表tに対するCREATE TABLE文を生成する。表tをスキーマ名とともにハッシュ表Bに登録する。表tをスキーマ名とともにハッシュ表Dに登録する。表tをオブジェクトとするGRANT文を下記のGRANT文生成手順で生成する。

(4) ここをLOOPラベルとする。

ハッシュ表Cを空にする。Cは、Dの要素であるビュー、実表を参照しているビューの全体を示す。ハッシュ表Dの各表dに対し、表dを参照している各ビューvに対し、以下a)を繰り返す。

a) もし、ビューvがスキーマ一覧Sの要素のいづれかのスキーマにあればビューvをハッシュ表Cに登録する。ハッシュ表Cが空であればDを空にして終り。そうでなければ、ハッシュ表Dを空にする。ハッシュ表Cの各表cに対して、以下a)、b)を繰り返す。ビューvが参照している各ビューまたは表tvに対し、

a) もしtvがハッシュ表Bにあればcontinue

b) もしtvがハッシュ表Bになければbreak

もしvが参照しているすべてのビューまたは実表がハッシュ表Bに存在すれば（つまりcontinueで上を終了すれば）、

a) vに関するCREATE VIEW文を生成する。

b) vをハッシュ表Bに登録する。

c) vをハッシュ表Dに登録する。

LOOPラベルに戻る。

【0102】以上の手順を図2に示す具体例を用いて説

明する。先ず図4に示されるようにハッシュ表B、C、Dが用意されているものとする。まず、スキーマS1から表T1が選択され、実表T1が生成され、実表T1がスキーマ名と共にハッシュ表Bとハッシュ表Dに登録される。そして、この実表T1をオブジェクトとするGRANTが生成される。ここまで時点では図2に示す68、69、70までの定義情報が生成される。次に、この実表T1を参照しているビューV1が存在しているので、これをハッシュ表Cに登録する。この時点でハッシュ表B、C、Dは図5に示すような状態になる。

【0103】次に、ハッシュ表Dが空にされ、ハッシュ表Cに登録されているビューV1が参照しているビューまたは実表（この場合には実表T1）が既にハッシュ表Bに登録されているため、ビューV1に関するCREATE文を生成する。この時点で図2の71が生成される。そして、ビューV1をハッシュ表B及びハッシュ表Dに登録する。この時点で各表は図6に示されるような状態になる。尚、図2の70については後述する。

【0104】次に、前述した手順のループラベルの部分に戻ることになる。ハッシュ表Cを空にし、ハッシュ表Dにある表V1に対して表V1を参照しているビューがあるかどうかをチェックする。この時点でビューV1を参照しているビューは存在していないため、ハッシュ表Cに登録するものは何もなく、ハッシュ表が空であるためこのループを終了する。

【0105】次にスキーマS2が持つ実表T2に対して同様の処理が行われる。図7に示すように、実表T2がスキーマ名と共にハッシュ表B及びハッシュ表Dに登録される。この時点で、各表は図7に示すような状態になる。次に、この実表T2に対するGRANT文を生成することになるが、この実表T2に対しては権限の付与がないため、何も生成されない。以上の時点までで、図2に示す72と73が生成されたことになる。

【0106】次にハッシュ表Cを空にし、ハッシュ表Dに登録された実表T2を参照しているビューがあるかどうかをチェックする。この場合にはビューV2が存在しているため、ビューV2をハッシュ表Cに登録する。この時点で各表は図8に示すような状態になる。

【0107】次にハッシュ表Dを空にし、ハッシュ表Cに登録されたビューV2が参照している各ビューまたは表が、ハッシュ表Bに登録されているかどうかをチェックする。ビューV2は実表T1とT2を参照しているが、このT1とT2は共に既にハッシュ表Bに登録されている。従ってビューV2に関するCREATE文を作成する。この時点で図2に示した74が生成される。そして、ビューV2をハッシュ表Bとハッシュ表Dに登録する。この時点で各表は図9に示したような状態になる。そしてループラベルに戻る。ループラベルに戻るとハッシュ表Cを空にし、ハッシュ表Dに登録されている表に対してその表を参照しているビューが有るかどうか

をチェックする。ビューV2に対しては、ビューV3が参照しているためビューV3をハッシュ表Cに登録する。この時点で各表は図10に示すような状態になる。ビューV3は、ビューV2のみを参照しており、このビューV2はすでにハッシュ表Bに登録されているため、このビューV3に対するCREATE文を生成する。この時点で図2の75が生成される。そしてビューV3をハッシュ表Bとハッシュ表Dに登録する。この時点で各表は図11に示すような状態になる。ここで再びループラベルに戻るが、次にはこのビューV3を参照するビューが存在しない為、ハッシュ表Cに新たに登録するビューは存在せず、ハッシュ表が空になるためこの処理を終了する。このようにして前述した手順を用いることにより、実表及びビューを矛盾なく生成することが可能になる。

【0108】次に、GRANT文生成の手順について説明する。

GRANT文生成の手順（入力：対象物名、所有者名）

- (1) ハッシュ表Gを空にする。
- (2) Gに"所有者名"を登録する。
- (3) ここをLOOPラベルとする。
- (4) Gが空ならばここで終わり。

そうでなければHを空にする。

G内の各人間gに対し、付与者がg、対象物が"対象物名"である権限すべてに関して、GRANT文を生成する。

権限がWITH GRANT OPTION付きであり、かつ権限受領者がユーザー名Uの要素であれば、Hにその権限の権限受領者を登録する。

G=Hとする。

LOOPラベルに戻る。

【0109】次に、上記GRANT文生成手順の具体例について説明する。このGRANT文生成手順は前述した実表、ビューの保存手順を生成する過程に呼ばれるものであり、ここでは図2に示した70のGRANT文が生成される場合について説明する。図2の70は\$U1が\$U2に対して選択権を与えていたものであるが、図12に示すように、ハッシュ表Gに先ず\$U1として所有者名鈴木が登録される。次にこの鈴木が持っている権限全てに関してGRANT文を生成する。このGRANT文の生成は、図24に示した権限一覧を参照することにより行われる。権限一覧にはシステムが権限を付与した場合と、特にユーザが他のユーザに権限を付与した場合が存在するが、ここでシステムが付与した権限は無視され、あるユーザが他のユーザに付与した権限のみが選択される。この例においてはユーザ鈴木がユーザ田中に選択権を与えていたため、ユーザ鈴木からユーザ田中に権限を再付与することができるユーザ名田中

を登録する。この時点で図13のような状態になる。次に、この表Hをハッシュ表Gに置き換える。即ち、図14のようにハッシュ表Gには田中が登録される。この田中が持つ権限を前述したと同様に権限一覧の中から検索し、もし田中が他のユーザに対して権限を与えている場合には、その権限全てに対して再びGRANT文が生成されることになるが、この例では田中が他のユーザに権限を与えていることは、権限一覧には登録されていないため、GRANT文は生成されない。このように、ユーザ1が鈴木である場合にはGRANT文が1つ生成される。

【0110】4) データベース及びカタログアクセス装置の動作

データベースの内容自身、及びカタログの内容をアクセスする装置である。この部分は、データベース中核に特

別なインターフェースを設けて実現することができるが、SQLなどでは、この部分自身もSQLで記述できる。まずデータベースのアクセスについて説明する。例えば、指定した実表T1の内容を取り出すには、

SELECT \* FROM T1

なるSQL文を実行すればよい。また、指定した内容を実表T1の内容にするには、

INSERT INTO T1

((1, AAA, 2), (2, bbb, 4), ...)

なるSQL文を実行すればよい。多くのSQLシステムでは更に高速の表の内容挿入インターフェースを持っているのでそれを用いてもよい。例えば、論理名' /DEV /MT00' なる磁気テープの内容を実表T1の内容とするには、

INSERT INTO T1

FROM /DEV/MT00

できる。

【0111】また、復元時及び移行時には、対象となるシステムにおけるカタログ内容を復元するには、上記のようにSQLの定義系コマンドを実行することにより自

動的に復元できる。例えば移行先で、

CREATE TABLE T1 (

t11 NUMERIC [4]

t12 CHARACTER [8]

を実行することにより、表一覧、列一覧情報中に所要の情報が設定される。また、表一覧情報中の表毎の列の個数なども保存内容になくともこれらコマンドの実行により自動的に復元される。あるいは、

CREATE VIEW V1 (v11, v12)  
AS SELECT \* FROM T1 WHERE t11 > 100

を実行することにより、ビュー表参照関係の情報や、スキーマ内に表が増加したことによるスキーマ一覧情報内の格納表数の列なども自動的に更新されることになる。即ち、これら情報は、退避の際に保存する必要がないことがわかる。

【0112】以上のように、この実施例は、データベースシステム中で開発された複数の業務から所要の業務に関連する定義情報を取り出し、これに加えて、業務に必要とされるデータ、業務プログラムのみをとりだし、業務パッケージとして作成し、これを別のマシン、システムに移行可能とすることを特徴とする。

【0113】また、定義情報の取りだしの際に、カタログ情報自身がSQLでアクセス可能な場合、カタログ情報の読みだしをSQLにより実行することにより特別なカタログ情報アクセス手段を不要とすることを特徴とする。

【0114】また、移行内容の内の定義情報の部分を、通常のSQLの定義情報定義のコマンド(CREATE

TABLEなど)の履歴的に矛盾のない列に変換し、復元時には特殊なシステムインターフェースを不要とし、通常のSQLの定義コマンド群を指定された順に実行することにより復元が可能とする。また、データ自身もSQLのデータ投入コマンドにより変換し、移行時これを実行することにより復元が可能になる。更に、これがSQLという国際標準言語を用いるため、ベンダを超えてパッケージの移行が可能となる。例えば、上記のスキーマ中の表の数などは、移行先で表定義のSQL文を順時実行することによりシステムにより通常の方法で自動的にこの部分を変更するため、関連する情報の移行が不要となる。また、特殊なインターフェースが不要となる。

【0115】また、ユーザ名、データが格納されるシステム上の位置など、移行先のシステムで衝突の可能性のあるものを置き換え可能とし、移行時にはこれらを適宜置き換えるだけでパッケージのインストールが可能なことを特徴とする。

【0116】また、SQLの手続き部分は保管せず、移

行先で実行を行なった際にカタログ内にこの手続きがないことによりその場で再コンパイルを行ない、その結果を登録することにより、SQLの手続き部を保管するための記憶を節約し、更に移行先での最適化パラメタなどに応じた業務プログラム移行を可能とすることを特徴とする。

【0117】また、データベース本体を退避／復元せず、復元時にデータベース定義及びその記憶領域のみを確保し、退避のための記憶領域を節約する。更に、これにより業務動作に必要な構組のみをパッケージとしてインストールし、データベース内容は移行先で業務の運用が進むに連れて蓄積されることを可能とする。

【0118】また、この手順を従来のデータベースの単純な保管／復元手順にも応用し、パッケージ生成とデータベースの保管／復元を一つの手順で実行することを特徴とする。

【0119】また、ジャーナルを保管する際には、ジャーナルの履歴レコードに表の識別子を付与しておくことによって、表単位の保管／復元を可能とすることを特徴とする。

【0120】実施例2. 次に、移行または復元の際には、退避されたデータをそのまますべて用いて、移行、または復元する場合に限らず、移行または復元の際に、退避移行復元手順生成装置3により移行先のシステム環境にあった移行仕様や、復元すべき対象物の限定等の復元仕様を与えることができる。例えば、復元する実表とそれを参照しているビューを復元するのではなく、復元時に実表に対してビューを評価実行し、その結果を実表として復元することで、以降先でのディスク容量を減らし、また用途に応じたデータのみの復元、移行を可能とする。

【0121】また、復元しようとする実表に対し、SQL文による選択、除外を実行し、移行先または復元先で必要なデータのみを復元することを可能とする。また、データベース本体を退避している場合でも、移行仕様を与えることにより、データベース定義及びその記憶領域のみを確保することも可能である。これにより業務動作に必要な構組のみをパッケージとしてインストールし、データベース内容は移行先で業務の運用が進むに連れて蓄積されることを可能とする。また、復元する場合でも、復元仕様を与えることにより一部の実表のみを復元することが可能になる。また、ジャーナルに対しても、保存されたジャーナルから復元する実表に対するジャーナルのみを選択することが可能になる。このように、退避されたデータの一部を用いて移行または復元する場合は、退避されたデータの中に依存情報が存在しているからであり、この依存情報も参照することにより、退避されたデータの中から更に、一部のデータを矛盾無く抽出することが可能になる。

【0122】実施例3. 上記実施例においてはSQLを

用いて説明したが、SQLを用いず他のデータベース処理言語を用いる場合においても、上記実施例と同様の効果を奏する。

【0123】実施例4. また、上記実施例1においてはデータベースシステム退避／移行／復元装置として、データベース及びプログラムの退避と移行と復元が1つの装置において出来る場合を説明したが、退避のみを行う装置、移行のみを行う装置、或いは復元のみを行う装置が別々に存在していても構わない。

【0124】

【発明の効果】以上のようにこの発明によれば、データベースシステムから指定した業務に必要な定義情報及びデータ及びプログラムを時間軸或いは相互関係に矛盾無く抽出することができ、これを用いてデータベースシステムを復元或いは移行する事が容易に出来るという効果がある。

【0125】また、データベースの復元とデータベースの移行を同一の仕様により行なうことができるため、データベースの復元と移行のための特別な処理手段を別々に必要としないという効果がある。

【0126】またデータベースシステムの退避をデータベース処理言語のコマンドの形に変換して退避するため、この退避したデータを復元する場合、或いは他のシステムに移行する場合には、データベース処理言語を実行できるシステムが備わっていればよく、特別な復元手段或いは、移行手段を必要としないという効果がある。

【図面の簡単な説明】

【図1】この発明に係るデータベース及びプログラム退避／移行／復元装置のブロック図を示す図である。

【図2】この発明に係るデータベースシステム退避装置により退避される保存情報の一例を示す図である。

【図3】この発明に係るデータベースシステムの実表とビューを退避する場合の退避すべき実表とビューの特定方法を示す図である。

【図4】この発明に係る退避手順生成装置の実表とビューの退避手順を説明する図である。

【図5】この発明に係る退避手順生成装置の実表とビューの退避手順を説明する図である。

【図6】この発明に係る退避手順生成装置の実表とビューの退避手順を説明する図である。

【図7】この発明に係る退避手順生成装置の実表とビューの退避手順を説明する図である。

【図8】この発明に係る退避手順生成装置の実表とビューの退避手順を説明する図である。

【図9】この発明に係る退避手順生成装置の実表とビューの退避手順を説明する図である。

【図10】この発明に係る退避手順生成装置の実表とビューの退避手順を説明する図である。

【図11】この発明に係る退避手順生成装置の実表とビューの退避手順を説明する図である。

【図12】この発明に係る退避手順生成装置がGRAN T文を生成する手順を説明するための図である。

【図13】この発明に係る退避手順生成装置がGRAN T文を生成する手順を説明するための図である。

【図14】この発明に係る退避手順生成装置がGRAN T文を生成する手順を説明するための図である。

【図15】従来のデータベースシステムのブロック図である。

【図16】従来のデータベースシステムのSQL手順の実行方式を説明するための図である。

【図17】データベースシステムのブロック図を示す図である。

【図18】ユーザー一覧情報の具体例を示す図である。

【図19】スキーマ一覧情報の具体例を示す図である。

【図20】表一覧情報の具体例を示す図である。

【図21】データベースシステムの列一覧情報を示す図である。

【図22】データベースシステムのビュー定義一覧を示す図である。

【図23】データベースシステムのビュー対表の参照関係を示す図である。

【図24】データベースシステムの権限一覧を示す図である。

【図25】データベースシステムのプログラム一覧を示す図である。

【図26】データベースシステムの手順一覧情報を示す図である。

【図27】データベースシステムの手順対表参照関係を示す図である。

【符号の説明】

- 1 コマンド解釈装置
- 2 制御部
- 3 退避移行復元手順生成装置
- 4 ビューSQL実行装置
- 5 保存媒体アクセス装置

6 データベース及びカタログアクセス装置  
 7 プログラムアクセス装置  
 8 保存媒体  
 9 データベースシステム  
 10 データベースシステム中核部  
 11 カタログ（定義情報）  
 12 データベース  
 13 ジャーナル  
 14 実行形式  
 【手続補正2】  
 【補正対象書類名】図面  
 【補正対象項目名】図10  
 【補正方法】変更  
 【補正内容】

【図10】

B 

S1. T1	S1. V1	S2. T2	S2. V2
--------	--------	--------	--------

C 

S3. V3
--------

D 

S2. V2
--------

【手続補正3】

【補正対象書類名】図面

【補正対象項目名】図19

【補正方法】変更

【補正内容】

【図19】

スキーマ一覧情報

スキーマ名	所有者	格納表数
S1	鈴木	2
S2	田中	2
S3	田中	1